#### https://www.halvorsen.blog



# Web Programming

Hans-Petter Halvorsen

### Contents

This tutorial gives a <u>brief overview</u> of web programming, HTML, CSS, JavaScript and different libraries and frameworks, both client-side (front-end) and server side (back-end).

- Introduction
  - Web, Internet, Web Technologies, etc.
- <u>HTML</u>
- <u>CSS</u>
  - <u>Bootstrap</u>
  - Icon Libraries
- JavaScript
- <u>Web Development Frameworks</u> (server side/back-end)
  - PHP, ASP.NET Core, Django, ...
- JavaScript Frameworks (mostly client-side/front-end)
  - Node.js, React, TypeScript, Angular, ...

### https://www.halvorsen.blog

Web Programming



# Introduction

#### Hans-Petter Halvorsen



## Web and Web Technology



# History of Internet and Web

- Internet (1960s)
- The first PC: 1981(IBM Personal Computer)
- World Wide Web WWW (1990)
- The first Web Browser Netscape, 1994
- Google, 1998
- Facebook, 2004
- Smartphones (iPhone), 2007
- Tablets (iPad), 2010
- Al and ChatGPT, 2022

# History of Internet and Web

- Internet (1960s)
  - Made it possible to connect computers all over the world
- The first PC: 1981(IBM Personal Computer)
  - Everyone could afford it and do it from home
- World Wide Web WWW (1990)
  - The foundation for todays web pages with hyperlinks
- The first Web Browser Netscape, 1994
- Google, 1998
  - Made it possible to search through all web pages and find information
- Facebook, 2004
  - A totally new way of using Internet and communicate with others
- Smartphones (iPhone), 2007
  - Internet and and browse web pages in your pocket
- Tablets (iPad), 2010
  - A new way of surfing and read information on Internet from your sofa
- Al and ChatGPT, 2022
  - Revolutionized the way of finding and creating information

### Internet and WWW

- The **Internet** and the World Wide Web (WWW) are the foundation of today's websites and web applications.
- The Internet and the WWW (often just called the web) are often used interchangeably, but they are different: **the Internet is the infrastructure**, while the **WWW is a service that runs on this infrastructure**.
- The internet is a global network that connects billions of devices, including computers, smartphones, and other digital devices. The Internet was founded in the 60s. The internet uses the **TCP/IP** protocol to send data between these devices.
- The WWW is a service that uses the internet to view and surf between web pages. The WWW was invented around 1990 by Tim Berners-Lee and made it possible to navigate between web pages using so-called hyperlinks.
- The WWW mainly uses the Hypertext Transfer Protocol (HTTP) to transfer information between clients (such as web browsers) and servers (so-called web servers).
- The first **web browsers** also came in the 90s. The first commercial browser was Netscape Navigator, which was launched in December 1994. This browser played an important role in popularizing the internet and making it accessible to a wider audience. Internet Explorer from Microsoft came along with Windows 95. Internet Explorer quickly became one of the most widely used browsers in the 1990s and early 2000s.
- **HTML** is the "language" used in connection with web pages. HTML is used to structure the content of web pages. Defined by Tim Berners-Lee in 1991.
- **CSS** (Cascading Style sheet) is used to format and present (layout) the web pages. Invented in 1994. This made it possible to separate content and presentation.
- JavaScript. Invented in 1995. Made it possible to create dynamic and interactive web pages.

# Web Programming Triangle



CSS - Cascading Style Sheets

## How Web Works



### **Computers and Network**



### URL

To retrieve a web page in the web browser, you must enter a web address or a so-called URL (short for "Uniform Resource Locator"). A URL is the address of a specific web page on the Internet. It is a unique web address that will lead you to a specific web page. Each website has its own unique URL/address. In the figure below we see the structure of such a URL:

#### https://blogg.usn.no/about/



### DNS

DNS is short for Domain Name System. DNS Servers translates domain names (like google.com) to IP addresses (like 142.250.74.110).



### Full stack Web Developers



# **Front-end Technologies**

Front-end "Programming Languages":

- HTML (HyperText Markup Language) You use HTML to define the content of web pages.
- CSS (Cascading Style Sheets) You use CSS to specify the layout of web pages.
- JavaScript You use JavaScript to program the behavior of web pages.

These 3 are the main building blocks for all Web Development.

# **Front-end Technologies**

- Front-end "Libraries and Frameworks":
- Bootstrap
- React
- Angular
- jQuery
- Vue

# **Back-end Technologies**

- PHP (PHP is both a framework and a programming language which is tailormade for web development)
- ASP.NET (C#)
- Django (Python)
- Node.js (JavaScript)
- +++

# Databases (Back-end)

- SQL Server
- MySQL
- MariaDB
- MongoDB
- +++ (we have hundreds of different database systems)

### References

• HTML Tutorial:

https://www.w3schools.com/html

• CSS Tutorial:

https://www.w3schools.com/css

• JavaScript Tutorial:

https://www.w3schools.com/js

### https://www.halvorsen.blog

#### Web Programming

Table of Contents



# HTML

#### You use HTML to define the content of web pages.

#### Hans-Petter Halvorsen



# Web Programming Triangle



CSS - Cascading Style Sheets

### HTML

- HTML is the main "language" used for web pages.
- HTML is used to structure the content of web pages.
- HTML stands for Hyper Text Markup Language
- HTML elements tell the browser how to display the content.

### HTML

<!DOCTYPE html> <html>

Here you see the main structure of an HTML file.

<head> <title>Hello World</title> </head>

<body>

<h1>Welcome</h1> Here you will learn about HTML.

</body> </html>

### Hello World

#### The HTML code created in Visual Studio Code:

•	$\leftarrow$ $\rightarrow$ $\bigcirc$ Search	
↔ hello_v	vorld.html ×	□ …
Users > h	<pre>halvorsen &gt; Library &gt; CloudStorage &gt; OneDrive-Personal &gt; Courses &gt; Webutvikling &gt; Tutorial:</pre>	s > HTML > Examples > <> hello_world.html > @ html The HTML page open in my local web browser:
3	choods	••• (a) (b) (c) (c) (c) (c) (c) (c) (c) (c) (c) (c
4 5	<pre><nead> <title>Hello World</title></nead></pre>	」> <title>Hello World</title> ← C 命 ① File   /Users/halvorsen/Library/CloudStorage/OneDrive-Per ☆ ③ 口 烇 @ ∞ …
6		🕐 Canvas 🛞 The Technical Guy 💪 Google 🐌 Copilot 🗢 Sikt KI-chat 🚳 ChatGPT 隆 Google Translate 🎦 Azure Devops 🎦 News
8	<body></body>	Welcome
9 10 11	<h1>Welcome</h1> Here you will learn about HTML	Here you will learn about HTML.
12 13		
14		Here, the HTML page is located on my PC and a
⑦ Restricted	Mode 🛞 0 🖄 0 👾 0	iust double-click to open the page in my default

You can use any type of text editor since HTML is pure text.

just double-click to open the page in my default web browser.

# **Upload File to Web Server**

#### You typically need to upload the HTML file to a Web Server using an FTP program:



# **HTML Tags Examples**

Hyperlink:

Here you see some of the most used HTML Tags

<a href="http://www.google.com">This is a link to Google</a>

**Bold Text:** 

Headers:

<h1>This is my Header</h1>

<h2>This is my Header</h2>

<h3>This is my Header</h3>

Title:

<title>This is my Title</title>

Paragraph:

My first paragraph.

Line Break: This is my Text <br>
This is also my Text

Comments:

<!-- Write your comments here -->

#### Image:

<img src="myimage.jpg" alt="blabla" width="104" height="142">

### Learn HTML

• HTML Tutorial:

https://www.w3schools.com/html

- HTML Fundamentals Videos (w3school): <u>https://www.youtube.com/playlist?list=PLP9IO</u> <u>4UYNF0VdAajP\_5pYG-jG2JRrG72s</u>
- HTML Tutorial:

https://www.geeksforgeeks.org/html-tutorial/

### https://www.halvorsen.blog

#### Web Programming

Table of Contents





#### You use CSS to specify the layout of web pages.

#### Hans-Petter Halvorsen



# Web Programming Triangle



### CSS

CSS (Cascading Style sheet) is used to format and present (layout) the web pages.



Ln 12, Col 25

# Three Ways to Insert CSS

There are three ways of inserting a style sheet:

#### Internal style sheet

- An internal style sheet should be used when a single document has a unique style.
- You define internal styles in the head section of an HTML page, inside the <style> tag.
- Inline style
  - An inline style loses many of the advantages of a style sheet (by mixing content with presentation). Use this method sparingly!
- External style sheet (Recommended!!)
  - An external style sheet is ideal when the style is applied to many pages. With an
    external style sheet, you can change the look of an entire Web site by changing just
    one file.
  - An external style sheet can be written in any text editor. The file should not contain any html tags.
  - The style sheet file must be saved with a .css extension.

### Internal CSS

Internal CSS is used to set the style for the entire HTML page. We then use the <style> element inside the <head> section. See Example:



### Inline CSS

For Inline CSS we use the style attribute for a single HTML tag. Examples:



## **External CSS**

#### stylesheet.css:

A recommended way to use CSS is to put all CSS styling and formatting into separate file and then link that document into the different HTML files. In that way you have all CSS in one place, and you can easily change the CSS for all your HTML files by only change it one place.

#### myfile.htm:

. . .

. . .

```
body {
    background-color: #d0e4fe;
}
h1 {
    color: orange;
    text-align: center;
}
p {
    font-family: "Times New Roman";
    font-size: 20px;
}
```

### Learn CSS

• CSS Tutorial:

https://www.w3schools.com/css

- CSS Fundamentals Videos (w3school): <u>https://www.youtube.com/playlist?list=PLP9</u> <u>IO4UYNF0UCaUSF3XNZ1U9f01E5h5PM</u>
- CSS Tutorial:

https://www.geeksforgeeks.org/css-tutorial

### https://www.halvorsen.blog

Web Programming



# Bootstrap

CSS (and JavaScript) Library

Hans-Petter Halvorsen



Table of Contents

### Bootstrap

- Bootstrap is the most popular front-end CSS Framework for developing responsive and mobile-first websites.
- Bootstrap is free to use.
- Homepage: <a href="https://getbootstrap.com">https://getbootstrap.com</a>
## **Using Bootstrap**

You can start using Bootstrap in different ways. The simplest method is to include the CSS and the JavaScript libraries in in the <head></head> section your HTML/PHP files.

#### 1. Put Bootstrap CSS Library:

khref="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet">

#### 2. Bootstrap JavaScript Library :

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js"></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></scri

#### 3. This line of code ensures your code works fine on all devices like smartphones, etc:

<meta name="viewport" content="width=device-width, initial-scale=1">

### Bootstrap - Hello World Example

• • •		$\leftarrow$ $\rightarrow$ (	, Search	<b>8</b> ~	
ф,	< bootstrap	_ex.html ×			□ …
_	Users > haiv	orsen > Library > CloudStorage > OneDrive-Personal > Courses > Webutvik	<pre>cling &gt; Tutorials &gt; HTML &gt; Examples &gt; &lt;&gt; bootstrap_ex.html &gt; 6</pre>	$\partial$ html $>$ $\bigcirc$ body $>$ $\oslash$ div.container-fluid $>$ $\oslash$ h	1
$\mathcal{P}$	1	html			
م	2	<html lang="en"></html>			
0	3				
¢>	4	<head></head>			
	5	<meta charset="utf-8"/>			
ш —	6	<meta content="&lt;/td" name="viewport"/> <td>"width=device-width, initial-</td> <td>scale=1"&gt;</td> <td></td>	"width=device-width, initial-	scale=1">	
-0	7	<pre><title>Bootstrap Example</title></pre>	e>		_
<b>P</b>	8	<link href="https://cdn.jsdeli&lt;/th&gt;&lt;th&gt;&lt;pre&gt;vr.net/npm/bootstrap@5.3.3/dis&lt;/pre&gt;&lt;/th&gt;&lt;th&gt;&lt;pre&gt;st/css/bootstrap.min.cs&lt;/pre&gt;&lt;/th&gt;&lt;th&gt;&lt;pre&gt;ss" rel="stylesheet"/>			
	9	<script src="https://cdn.jsdel&lt;/th&gt;&lt;th&gt;&lt;pre&gt;ivr.net/npm/bootstrap@5.3.3/d&lt;/pre&gt;&lt;/th&gt;&lt;th&gt;ist/js/bootstrap.bundle&lt;/th&gt;&lt;th&gt;e.min.js"></script>			
	10				
	11				
	12	<body></body>			
	13	<pre><div class="container-fluid"></div></pre>			
	14	<h1>Hello World!</h1>			
	15				
	16				
8	17				
£223	18				

### Example

6 👩	▶ 1 🔽   🖎   🕅 🕒 Books App 🛛 🗙 + - □ ×									
$\leftarrow$ (	← C 🞧 🗈 https://web01.usn.no/~hansha/books.php 🖒 <♪ <≧ … 🥠									
H The Tee	chnical Guy 🕒	Google 🧕 Goo	gle Translate 🛛 b Micro	osoft Copilot 🛛 🚳 Cha	atGPT > 🗋 Other	favourites				
Boo	ks							E	Bootst	rap
Here you	find a list of av	vailable books:			□   ①   ℕ   Ĥ [	Books App	× +	\$	- □ ) <u>{</u>	×
BookId	Title	Author	Topic	H The Technical Guy	G Google 隆 Google Translate	b Microsoft Cop	vilot 💿 ChatGPT 🎦 News 🎦 A	Adm >	C Other fav	vourites
1	Web Apps	Elvis Presly	Programming							
2	IoT and Cloud	John Wayne	IoT	Pooks						
3	C#	Rune Hansen	Programming	DOOKS						
4	AI	Allan Johnsen	Data	Here you find a lis	st of available books:					
				BookId	Title	Au	ithor	Торіс		
				1	Web Apps	Elv	is Presly	Programming		
				2	IoT and Cloud	lot	nn Wayne	IoT		
Plain	HTML			3	C#	Ru	ne Hansen	Programming		
				4	AI	All	an Johnsen	Data		

### https://www.halvorsen.blog

Web Programming



### Hans-Petter Halvorsen



### **Icon Libraries**

When creating web pages, you typically want to include icons. Here are some examples of different Icon libraries that can be used:

- Google Fonts
  - Free to use
  - https://fonts.google.com/icons
- Bootstrap
  - Free to use
  - <u>https://icons.getbootstrap.com</u>
- Font Awesome
  - Free for small projects
  - https://fontawesome.com

### **Google Fonts**

3	C	🖁 Reset all 🗙	🄏 Google	Fonts Q	Search icons						< >
its	∃≟ Customize		× Eiltere							Check Circl	e C
to	Fill (j		A Pilters								Size - 24 +
ns ≩1	Weight (j)		Material Desi guidelines Learn about the	gn icon	Figma plugin Use the Material Syn plugin on Figma	mbols	GitHub repo This repository contr the binary font file se	ains erved	Apache license Material Symbols are available under the	0	Color
edge	100	700	best practices fo	oricons			by Google Fonts		Apache License Version 2.0	Ľ	<u>◆</u> #1F1F1F
Q	Grade (j)		UI actions							🛃 svg	y PNG
	-25 (low)	200 (high emphasis)	Q	ŵ	$\equiv$	×	<b>\$</b>	$\odot$	$\heartsuit$	Web	Android Apple
	Optical Size 🥡		Search	Home	Menu	Close	Settings	Check Circle	Favorite	Instructions	
	20рх	48px	$\leftarrow$	\$	>	[→	>	(+)	$\otimes$	Check the Materia advanced exampl and font loading o	al Symbols guide for es such as animations optimization.
	Filter		Arrow Back	Star	Chevron Right	Logout	Arrow Forward iOS	Add Circle	Cancel	Variable icon font Add the variable f your head tag and	t ont stylesheet request t I the current variable ax
	압 Style	^	•		$\checkmark$		[7]		C	configuration to i	cons using CSS.
	Material Symbols	(new) -	Arrow Drop Down	More Vert	Check	Check Box	Open In New	Toggle On	Refresh	<link href="htt&lt;br&gt;ogleapis.com/css2?f&lt;br&gt;l+Symbols+Outlined:&lt;br&gt;L,GRAD@2048,100&lt;br&gt;.200&amp;icon_names=che&lt;/td&gt;&lt;/tr&gt;&lt;tr&gt;&lt;td&gt;&lt;/td&gt;&lt;td&gt;outined&lt;/td&gt;&lt;td&gt;&lt;/td&gt;&lt;td&gt;&lt;&lt;/td&gt;&lt;td&gt;&lt;math&gt;\bigcirc&lt;/math&gt;&lt;/td&gt;&lt;td&gt;&lt;/td&gt;&lt;td&gt;⊻&lt;/td&gt;&lt;td&gt;0 0 0&lt;br&gt;0 0 0&lt;br&gt;0 0 0&lt;/td&gt;&lt;td&gt;&lt;math&gt;\rightarrow&lt;/math&gt;&lt;/td&gt;&lt;td&gt;۲&lt;/td&gt;&lt;td&gt;ck_circle" rel="st&lt;br&gt;ps://fonts.go&lt;br&gt;amily=Materia&lt;br&gt;opsz,wght,FIL&lt;br&gt;700,01,-50.&lt;/td&gt;&lt;td&gt;ylesheet"/>	Copy code
	Category	^	Chevron Left	Radio Button Unchecked	More Horiz	Download	Apps	Arrow Right Alt	Radio Button Checked	<style></style>	

https://fonts.google.com/icons

Home

Household

### https://www.halvorsen.blog

Web Programming

Table of Contents



# JavaScript

You use JavaScript to program the behavior of web pages.

### Hans-Petter Halvorsen

## Web Programming Triangle



- JavaScript is the programming language of the Web.
- All modern HTML pages are using JavaScript.
- JavaScript is the default scripting language in all modern browsers, and in HTML5. JavaScript is free to use.
- JavaScript is probably the most popular programming language in the world.
- It is the language for HTML, for the Web, for computers, servers, laptops, tablets, smart phones, and more.
- JavaScript can change HTML Elements! which makes it very powerful!
- You don't need to download or install JavaScript. JavaScript is already running in your browser on your computer, on your tablet, and on your smart-phone.

- JavaScript has all the features that other programming languages have like variables, for loops, while loops, if-else, functions, events, arrays, etc.
- In addition, it has many HTML specific features
  - Like reading or changing HTML elements.

- You use JavaScript to program the behavior of web pages.
- You can use JavaScript to make your web pages more dynamic, e.g.,:
  - you can specify what happens when a user clicks on a button, etc.
  - You can update contents on the HTML web page.
  - You can create and use functions, create variables, if else, etc. – just like other programming languages.
- You use the <script> tag to define JavaScript code within your HTML files.

### Variables

```
<!DOCTYPE html>
<html>
<head>
    <title></title>
</head>
<script>
  let number = 12;
  let title = "Hello World";
  const name = "Hans-Petter";
</script>
<body>
</body>
</html>
```

#### You use **let** to declare variables





### JavaScript Example

### Calculator

Enter values for the Numbers and JavaScript will Calculate the Sum:



## document.getElementById()

- document.getElementById("id").innerHTML
  - Use innerHTML when you want to change an HTML element.
- document.getElementById("id").innerText
  - Use innerText when you only want to change the plain text.
- document.getElementById("id").value
  - Use **value** when you get/set data from textboxes, etc.

•	0	Calculate	n	×	+		-		$\times$
~	$\rightarrow$	C	(i) File	C:/Temp/Web/	calculator.html (	Ð	☆	2	:

#### JavaScript Example

#### Calculator

Enter values for the Numbers and JavaScript will Calculate the Sum:

3	+	4	:	=	7
					L

```
<script>
function Add(){
```

```
let number1 = document.getElementById("number1").value;
number1 =parseFloat(number1);
```

```
let number2 = document.getElementById("number2").value;
number2 =parseFloat(number2);
```

```
let sum = number1 + number2;
document.getElementById("answer").value = sum;
//alert(sum);
```

```
</script>
```

}

```
<body>
<h1>JavaScript Example</h1>
```

```
<h1>Calculator</h1>
```

Enter values for the Numbers and JavaScript will Calculate the Sum:

```
<input type="number" id="number1">
+
<input type="number" id="number2">
<button onclick="Add()">=</button>
<input type="number" id="answer">
</body>
```

</html>

### **Button Click Example**

ers > halvorsen > Libra	y > CloudStorage > OneDrive-Personal > Courses > Webutvikling > Tutorials > HTML > Examp	oles > 💠 button_click_ex.html > 🔗 html			
1 DOCT</td <td>YPE html&gt;</td> <td></td> <td>_</td> <td></td> <td></td>	YPE html>		_		
2 <html></html>					
3 <head></head>					
4 <t< td=""><td><pre>itle&gt;Button Click Example</pre></td><td></td><td></td><td></td><td></td></t<>	<pre>itle&gt;Button Click Example</pre>				
5 <5	cript>				
6	<pre>function ButtonClick() {</pre>				
/	alert("The author of this article is	Hans-Petter Halvorsen")	;		
0	}	••• <b>a</b> m	Button Click Exam	nole x +	
10 <td>&gt;</td> <td>○ A C →</td> <td>File /Users/halvorsen,</td> <td>/Library/CloudStorage/OneDrive-Personal/Courses/ ☆</td> <td>\$ 0 4 6 %</td>	>	○ A C →	File /Users/halvorsen,	/Library/CloudStorage/OneDrive-Personal/Courses/ ☆	\$ 0 4 6 %
		🔘 Canvas 🛛 🛞 The Techni	cal Guy G Google Thi	s page says	ure Devops 🕒 News
II Souge	1>Introduction	Introduction	The	e author of this article is Hans-Petter Halvorsen	
12 <h< td=""><td></td><td></td><td></td><td>OK</td><td></td></h<>				OK	
12 <h< td=""><td>Here you will learn about HTML.</td><td>Here you will learn about</td><td>HTML.</td><td></td><td>_</td></h<>	Here you will learn about HTML.	Here you will learn about	HTML.		_
12 <r 13="" 14<="" td=""><td>Here you will learn about HTML.</td><td>Here you will learn about Show Author</td><td>HTML.</td><td></td><td></td></r>	Here you will learn about HTML.	Here you will learn about Show Author	HTML.		
12 <t 13 14 15 12</t 	<pre>Here you will learn about HTML. <button onclick="ButtonClick()">Show Auth</button></pre>	Or	HTML.		
12 <t 13 14 15 16</t 	<pre>Here you will learn about HTML. <button onclick="ButtonClick()">Show Auth</button></pre>	or Here you will learn about Show Author Background Here you will learn about	HTML.	www.	
12 <t 13 14 15 16 17 17 17 17 17 17 17 17 17 17 17 17 17</t 	<pre>Here you will learn about HTML. <button onclick="ButtonClick()">Show Auth <h2>Background</h2></button></pre>	or Here you will learn about Show Author Background Here you will learn about	HTML.	www.	
12 <r< td="">       13     14       15     16       17     18</r<>	<pre>Here you will learn about HTML. <button onclick="ButtonClick()">Show Auth <h2>Background</h2> Here you will learn about the history</button></pre>	or Here you will learn about Show Author Background Here you will learn about	HTML.	www.	
12 <t 13 14 15 16 17 18 19 <td><pre>Here you will learn about HTML. <button onclick="ButtonClick()">Show Auth <h2>Background</h2> Here you will learn about the history &gt;</button></pre></td><td>or Here you will learn about Show Author Background Here you will learn about</td><td>HTML.</td><td>www.</td><td></td></t 	<pre>Here you will learn about HTML. <button onclick="ButtonClick()">Show Auth <h2>Background</h2> Here you will learn about the history &gt;</button></pre>	or Here you will learn about Show Author Background Here you will learn about	HTML.	www.	

### Update Web contents

🗕 🔍 🌒 🍘 🗖 🕒 Button Click Exa	mple × +						
🔶 🖒 🕕 File   /Users/halvorsen/Library/CloudStorage/OneDrive-Personal/Courses/Webutvi 🏠 🕄 🗇 🗲 🔂 😵 … 🥠							
🕐 Canvas 🛞 The Technical Guy 🔓 Google 🐌	This page says	vevops 🗀 News					
Introduction	Write your Name Hans-Petter						
Here you will learn about HTML.	Cancel	ОК					
Background		Introduction					
Here you will learn about the history of Internet and	1 WWW.	Here you will learn about HTML.					
Here we use Jay update the cont	vaScript to tents of a	Hans-Petter					
"textbox" based	d on user input.	Background					
		Here you will learn about the history of Internet and WWW.					

#### Update Web contents • • • textbox ex.html × .... ப Users > halvorsen > Library > CloudStorage > OneDrive-Personal > Courses > Webutvikling > Tutorials > HTML > Examples > 📀 textbox\_ex.html > 🔗 html > 🔗 head > 🔗 script <!DOCTYPE html> Button Click Example × + 2 <html> /Users/halvorsen/Library/CloudStorage/OneDrive-Personal/Courses/Webutvi. 3 <head> This page says Write your Name <title>Button Click Example</title> 4 Introduction Hans-Petter script> 5 Here you will learn about HTML Write your Name function ButtonWriteNameClick() { 6 Write Name Γø 7 let myname; Background Here you will learn about the history of Internet and WWW 8 myname = prompt("Write your Name"); 9 Ē if (myname != null && myname != "") 10 document.getElementById("myname").innerHTML = myname 11 12 13 </script> </head> 14 15 <body> <h1>Introduction</h1> 16 Here you will learn about HTML. 17 18 <textarea id="myname">Write your Name</textarea> 19 20 <br>br>21 <button onclick="ButtonWriteNameClick()">Write Name</button> 22 23 <h2>Background</h2> 24 Here you will learn about the history of Internet and WWW. </body> 25 </html> 26 🐨 Restricted Mode 🛛 🛇 🛆 0 👷 0 Ln 13, Col 14 Spaces: 4 UTF-8 LF HTML

## Learn JavaScript

- JavaScript Tutorial: <u>https://www.w3schools.com/js</u>
- JavaScript Fundamentals Videos (w3school): <u>https://www.youtube.com/playlist?list=PLP9IO4UY</u> <u>NF0WWmZpE3W33vVPRl2GvjEqz</u>
- The Modern JavaScript Tutorial: <u>https://javascript.info</u>
- JavaScript Tutorial: <u>https://www.geeksforgeeks.org/javascript/</u>

**Table of Contents** 



# Web Programming Web Development Frameworks

Server-side (backend) Web Development Frameworks

Hans-Petter Halvorsen

## Web Development Frameworks

Some of the most used server-side (backend) Web Development Frameworks:

- PHP (Scripting language for Web development)
- ASP.NET (Microsoft, Visual Studio, C#)
- Ruby on Rails (Ruby)
- Django (Python)
- +++

### Web Development Frameworks

What is a <u>server-side</u> Web Development Framework?

- It is used to generate <u>dynamic</u> web page content before the page is sent to the user's web browser.
- With dynamic we typically mean that the web page/application is dynamically changed based on data in a Database, etc.

### How it Works



### PHP



- PHP is a server scripting language for making dynamic and interactive web pages.
- PHP scripts are executed on the server/webserver.
- PHP files have extension ".php" and are typically a mix of PHP, HTML, CSS and JavaScript
- PHP is free and open-source.
- With PHP you can easily typically. communicate with a Database, and especially MySQL.
- LAMP: Linux, Apache, MySQL and PHP.
- PHP is widely used and still by far the most used/popular language for web development.
- PHP is easy to learn (but still very powerful) which cannot be said on many other web technologies and programming languages.
- Homepage: <u>https://www.php.net</u>
- PHP Tutorial: <a href="https://www.w3schools.com/php">https://www.w3schools.com/php</a>

### **PHP** Example

Typically, you include PHP code in between your HTML code. Here is a basic example:



Hello World

Hello, my name is Hans-Petter Halvorsen

```
<!DOCTYPE html>
<html>
    <body>
        <h1>Hello World</h1>
        <?php
            $name ="Hans-Petter Halvorsen";
            echo "Hello, my name is $name";
        ?>
    </body>
</html>
```

PHP Code is put inside <?php ... ?> and it is executed on the server-side before it is sent to the web browser on the client side.

### **ASP.NET** with Razor

- ASP.NET is a framework for web development.
- ASP.NET is based on .NET and C#.
- What is the difference between ASP.NET and .NET frameworks?
  - ASP.NET is specifically designed for web development, while the .NET framework covers a broader range of application types, including Windows desktop, mobile, and web applications.
- In ASP.NET code and layout are separated into 2 files; The layout file has the extension ". cshtml", and the code-behind file has the extension ". cshtml.cs" (where "cs" is short for C#).
- The layout file ". cshtml" use something called Razor syntax and are mixed with HTML.
- Homepage: <a href="https://dotnet.microsoft.com/en-us/apps/aspnet">https://dotnet.microsoft.com/en-us/apps/aspnet</a>

### **ASP.NET** with Razor

- Razor is a markup syntax for embedding server-based code into ASP.NET Core webpages.
- The Razor syntax consists of Razor markup, C#, and HTML.
- Files containing Razor generally have a .cshtml file extension.
- The default Razor language is HTML.
- Rendering HTML from Razor markup is no different than rendering HTML from an HTML file.
- HTML markup in .cshtml Razor files is rendered by the server unchanged.

### PHP vs ASP.NET

- PHP is open-source and free to use.
- With ASP.NET you are locked and forced into Microsoft technology and tools.
- PHP works on various platforms and supports most web servers.
- ASP.NET can have a steeper learning curve for beginners.
- PHP is embedded within the HTML code, making web development easier.
- Hosting is very easy with PHP with lots of providers.
- ASP.NET is based on C#, .NET and Visual Studio, so if you already know these tools, then you are well on your way.

https://eluminoustechnologies.com/blog/asp-net-vs-php/

### **ASP.NET YouTube**

 Back-end Web Development with .NET for Beginners:

https://www.youtube.com/playlist?list=PLdo4f OcmZ0oWunQnm3WnZxJrseIw2zSAk

 Front-end Web Development with .NET for Beginners:

https://www.youtube.com/playlist?list=PLdo4f OcmZ0oXNZX1Q8rB-5xgTSKR8qA5k



Python has become very popular the last 10 years and can also be used for Web development.

Python Frameworks for web development:

- Django
- Flask

# Django (Python)

- Django is a back-end (server-side) web development framework.
- Django is free, open source and written in Python.
- Django is using a Model View Template (MVT) pattern.
- Homepage: <a href="https://www.djangoproject.com">https://www.djangoproject.com</a>
- Django Tutorial: <u>https://www.w3schools.com/django</u>

## Flask (Python)

- Flask is a "micro" web framework written in Python.
- Flask can be used to build lightweight web applications, you can run it on, e.g., Raspberry Pi.
- Wikipedia:

https://en.wikipedia.org/wiki/Flask\_(web\_framework)

- Homepage: <a href="https://flask.palletsprojects.com">https://flask.palletsprojects.com</a>
- Flask Tutorial: <u>https://www.geeksforgeeks.org/flask-</u> <u>tutorial/</u>

# Web Programming JavaScript Libraries and Frameworks

Client-side (Frontend) Frameworks and Libraries based on JavaScript or similar

Hans-Petter Halvorsen

**Table of Contents** 

### JavaScript Libraries and Frameworks

- JavaScript is a programming language.
- JavaScript is probably the most used programming language in the world.
- JavaScript is an open standard that is maintained by Ecma International and the W3C.
- All web browsers has built-in support for JavaScript.
- The last 10-15 years lots of new Libraries or Frameworks has been created that is based on JavaScript.
- Why? These Libraries or Frameworks builds functionality on top of JavaScript to make it easier to create more advanced web applications with more complex GUI, etc.
- The problem?
  - It has been made quit a lot of these Libraries or Frameworks, so it has become difficult to choose between them.
  - These Libraries and Frameworks have also some overhead, installation issues, etc. compared to original JavaScript.

### How it Works


#### JavaScript Libraries and Frameworks

- "Vanilla" JavaScript which is just plain JavaScript 🙂
  - People use "Vanilla" JavaScript as a joke to remind other developers that many things can be done without the need for additional JavaScript libraries.
  - I guess it is also important to learn the fundamentals in plain JavaScript before starting to use other frameworks that is based on JavaScript.
- Node.js
- **TypeScript** (originally made by Microsoft)
  - TypeScript is a superset of JavaScript, meaning it builds on JavaScript by adding static type definitions.
- React (originally made by Facebook)
- Angular (originally made by Google)
- jQuery
- Vue
- +++

# Node.js

- Node.js allows you to run JavaScript on the server (server-side, backend).
- Node.js is a free, open-source and cross-platform
- npm is a package manager for Node.js packages/modules
- Homepage: <u>https://nodejs.org</u>
- Node.js Tutorial: <u>https://www.w3schools.com/nodejs</u>
- npm is used to install many other frameworks/libraries like TypeScript, React, etc.

#### Install Node.js



### Node.js



After installation of Node.js you can open the Command Prompt to confirm that Node has been successfully installed. If you enter "node -v in", the installed version of Node.js will be shown.

#### Node.js in Visual Studio

2 8 - D X



#### https://visualstudio.microsoft.com/vs/features/node-js/

## NMP and NPX

- **NPM** is a package manager, and it is installed as part of Node.js.
- NPM is a package manager used to install, delete, and update JavaScript packages on your machine.
- NPM is short for Node Package Manager.
- NPX is a package executer, and it is used to execute JavaScript packages directly, without installing them.
- NPX is short for Node Package eXecute.

# NPM

- npm is the world's largest Software Library.
  - Homepage: <u>https://www.npmjs.com</u>
- npm is also a Software Package Manager and Installer.
- npm is free to use.
- npm includes a CLI (Command Line Client) that can be used to download and install software:
  - C:\>npm install <package>
- The name npm (Node Package Manager) comes from Node.js since it was originally created as a package manager for Node.js.
- For most of the JavaScript packages you need to use npm to install them.
- npm is installed with Node.js
  - This means that you must install Node.js to get npm installed on your computer.
  - Download: <a href="https://nodejs.org">https://nodejs.org</a>

# TypeScript

- TS
- Microsoft created TypeScript in 2012 to address shortcomings in JavaScript.
- TypeScript is a strongly typed programming language that builds on JavaScript.
  - For example, TypeScript will report an error when passing a string into a function that expects a number. JavaScript will not.
- TypeScript is a superset of JavaScript and adds syntax on top of JavaScript.
- Homepage: <u>https://www.typescriptlang.org</u>
- TypeScript Tutorial: <u>https://www.w3schools.com/typescript</u>

# TypeScript

How to start using TypeScript:

Use TypeScript online with TypeScript
 Playground:

https://www.typescriptlang.org/play

- Download TypeScript using **npm**.
- Download and start using TypeScript using Visual Studio.

#### React



- The React library is an open-source JavaScript library you can use to create dynamic and appealing applications with rich UI (User Interfaces).
- Basically, React is a front-end library for creating UI.

#### React



- React is a front-end JavaScript library for building user interfaces in Web Applications.
- React is also known as "React.js" or "ReactJS".
- Developed by Facebook in 2011. React is now free and open source.
- React mainly focus on the user interface and rendering components to the DOM.
- Note! React is a "front-end Library" and not a "Framework".
  - So, you typically need a Framework like Next.js or ASP.NET Core, etc.
- Homepage: <u>https://react.dev</u>.
- React Quick Start: <a href="https://react.dev/learn">https://react.dev/learn</a>
- React Tutorial: <u>https://www.w3schools.com/react/</u>.
- Tutorial: Create an ASP.NET Core app with React in Visual Studio: https://learn.microsoft.com/en-us/visualstudio/javascript/tutorial-aspnet-core-with-react.

#### Test out React – Hello World

$\leftarrow \rightarrow$	C 🛈 File	C:/Temp/Web/	'test_react.html	Q	☆	2	:
He	llo, y	worl	d!				
	,						

- To use React in production, you need npm which is included with Node.js.
- To start learning React you can write React directly in your HTML files, as you see in this example.

File Edit Selection	View Go Run Terminal Help         ← → <th< th="">           &lt;</th<>
<pre>     test_react.html ×     C: &gt; Temp &gt; Web &gt; </pre>	<     O test react.html ) Ø html ) Ø bodv ) Ø snipt
1 <	<pre><!DOCTYPE html>    </pre>
2 <	<pre>chtml&gt;</pre>
3	<head></head>
4	<pre><meta charset="utf-8"/> Vous poor of the implication of ODN loss</pre>
5	<pre><title>Hello World</title> YOU NEED to INCLUDE THESE 3 CDINS:</pre>
6	<pre><script src="https://unpkg.com/react@18/umd/react.development.js"></script></pre>
7	<pre><script src="https://unpkg.com/react-dom@18/umd/react-dom.development.js"></script></pre>
8	<pre><script src="https://unpkg.com/@babel/standalone/babel.min.js"></script></pre>
9	
10	<body></body>
11	<div id="root"></div>
12	<script type="text/babel"></td></tr><tr><td>13</td><td></td></tr><tr><td>14</td><td><pre>function MyApp() {</pre></td></tr><tr><td>15</td><td><pre>return <h1>Hello, world!</h1>;</pre></td></tr><tr><td>16</td><td>}</td></tr><tr><td>17</td><td></td></tr><tr><td>18</td><td><pre>const container = document.getElementById('root');</pre></td></tr><tr><td>19</td><td><pre>const root = ReactDOM.createRoot(container);</pre></td></tr><tr><td>20</td><td><pre>root.render(<MyApp />);</pre></td></tr><tr><td>21</td><td></td></tr><tr><td>22</td><td></script>
23	
3 24 <b>&lt;</b>	<pre>content Delivery Network (CDN)</pre>

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8" />
    <title>Hello World</title>
    <script src="https://unpkg.com/react@18/umd/react.development.js"></script></script></script></script>
    <script src="https://unpkg.com/react-dom@18/umd/react-dom.development.js"></script>
    <script src="https://unpkg.com/@babel/standalone/babel.min.js"></script></script></script></script>
  </head>
  <body>
    <div id="root"></div>
    <script type="text/babel">
      function MyApp() {
         return <h1>Hello, world!</h1>;
      }
      const container = document.getElementById('root');
      const root = ReactDOM.createRoot(container);
      root.render(<MyApp />);
```

</script> </body> </html>

#### **Install React**

- To start learning React you can write React directly in your HTML files as shown in the example.
- You can create a React App like this (using CRA): npx create-react-app myreactapp
- Or you can use a build tool like Vite.
- Here you find more information about installation and creating React Apps <u>https://react.dev/learn/installation</u>
- How to Install React A Step-by-Step Guide: <u>https://www.freecodecamp.org/news/how-to-install-react-a-step-by-step-guide/</u>

## React Components

- In React you create and use Components.
- You create and use Components to build your User Interfaces (UI).
- In a React app, every piece of UI is a Component.
- Components are independent and reusable bits of code.
- Basically, Components are reusable UI elements for your React Web App.
- They serve the same purpose as JavaScript functions but work in isolation and return HTML.
- React Components are regular JavaScript functions, but their names must start with a capital letter, or they will not work!

#### **Component Example**

🗙 File	Edit Selection	n View Go Run Terminal Help ← → $\rho$ Search &	
Ø	react_component	nt_ex.html ×	
ρ	C: > Temp > Web . 1	<pre>/ Component_extrain &gt; @ mmi &gt; @ booy &gt; @ script / DOCTYPE html&gt;</pre>	
مع	2	<html></html>	
	3	<head></head>	
 	4	<meta charset="utf-8"/>	
	5	<title>Hello World</title>	
	6	<pre><script src="https://unpkg.com/react@18/umd/react.devel&lt;/pre&gt;&lt;/th&gt;&lt;th&gt;opment.js"></script></pre>	
	7	<pre><script src="https://unpkg.com/react-dom@18/umd/react-d&lt;/pre&gt;&lt;/th&gt;&lt;th&gt;om.development.js"></script></pre>	
	8	<pre><script src="https://unpkg.com/@babel/standalone/babel.&lt;/pre&gt;&lt;/th&gt;&lt;th&gt;min.js"></script></pre>	
	9		
	10	<body></body>	
	11	<h1>React Components</h1>	
	12	<div id="root"></div>	✓ S Hello World × Hello World × + - □ ×
	13	<script type="text/babel"></th><th></th></tr><tr><th></th><th>14</th><th></th><th>← → C ① File C:/Temp/Web/react_component_ex.html @ ☆ @ :</th></tr><tr><th></th><th>15</th><th><pre>function Button() {</pre></th><th></th></tr><tr><th></th><th>16</th><th><pre>return <button>Click Me</button>;</pre></th><th></th></tr><tr><th></th><th>17</th><th>}</th><th></th></tr><tr><th></th><th>18</th><th></th><th>React Components</th></tr><tr><th></th><th>19</th><th><pre>const container = document.getElementById('root');</pre></th><th>React Components</th></tr><tr><th></th><th>20</th><th><pre>const root = ReactDOM.createRoot(container);</pre></th><th></th></tr><tr><th></th><th>21</th><th><pre>root.render(<Button />);</pre></th><th>Oliak Ma</th></tr><tr><th></th><th>22</th><th></th><th></th></tr><tr><th></th><th>23</th><th></script>	
8	24		
£53	25		

Restricted Mode

# React Props

- Props (or Properties) are arguments passed into React Components.
- Props are passed to Components as HTML Attributes.
- Props are like function arguments in JavaScript and attributes in HTML.

#### Props Example

File	e Edit Selection	n View (	Go Run Terminal Help	$\leftrightarrow$ $\rightarrow$	, ○ Search								
<b>,</b>	• react_compone	ent_props_e	ex.html ×						• ••				
	C: > Temp > Web	> • react	t_component_props_ex.html > ∲ html > ∳	≥ body > 😌 script > 💬 Button									
	1	D0</td <td>OCTYPE html&gt;</td> <td></td>	OCTYPE html>										
9	2	<htr< td=""><td>ml&gt;</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></htr<>	ml>										
>	3	<	head>										
R	4		<meta charset<="" td=""/> <td>="UTF-8" /&gt;</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td>	="UTF-8" />									
	5		<title>Hello</title>	World									
	6		<script src="&lt;/td&gt;&lt;td&gt;https://unpkg.c&lt;/td&gt;&lt;td&gt;om/react@18/umd/rea&lt;/td&gt;&lt;td&gt;ct.developmer&lt;/td&gt;&lt;td&gt;t.js"></scr</td><td>ipt></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td>7</td><td></td><td><script src="</td><td>https://unpkg.c</td><td>om/react-dom@18/umd</td><td>/react-dom.de</td><td>velopment.j</td><td><mark>s</mark>"></scr</td><td>ipt></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td>8</td><td></td><td><script src="</td><td>https://unpkg.c</td><td>om/@babel/standalon</td><td>e/babel.min.j</td><td><mark>s</mark>"></script</td><td>></td><td>_</td><td>_</td><td></td><td></td><td></td></tr><tr><td></td><td>9</td><td><,</td><td>/head></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td>10</td><td><t</td><td>body></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td>11</td><td></td><td><h1>React Com</td><td>ponents</h1></td><td></td><td></td><td>-</td><td></td><td></td><td></td><td></td><td></td><td>_</td></tr><tr><td></td><td>12</td><td></td><td><div id="root</td><td>"></div></td><td></td><td><math>\sim</math></td><td>🕙 Hello World</td><td></td><td>×</td><td>Hello World</td><td>× +</td><td>-</td><td></td></tr><tr><td></td><td>13</td><td></td><td><script type=</td><td>"text/babel"></td><td></td><td></td><td></td><td>File Cu/Terr</td><td>an (Mah (</td><td>wast component p</td><td>rans ov html</td><td>@ ~</td><td></td></tr><tr><td></td><td>14</td><td></td><td></td><td></td><td></td><td>~</td><td>- 0 0</td><td>rile C./Ten</td><td>ip/web/</td><td>react_component_p</td><td>nops_ex.num</td><td>~ ×</td><td></td></tr><tr><td></td><td>15</td><td></td><td>function Bu</td><td><pre>tton(props) {</pre></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td>16</td><td></td><td>return <b</td><td>utton>{props.na</td><td><pre>me}</button>;</pre></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td>17</td><td></td><td>}</td><td></td><td></td><td></td><td></td><td></td><td>۲</td><td></td><td></td><td></td><td></td></tr><tr><td></td><td>18</td><td></td><td></td><td></td><td></td><td></td><td>keac</td><td>PT (</td><td><math>\mathbf{O}</math></td><td>mno</td><td>nent</td><td>S</td><td></td></tr><tr><td></td><td>19</td><td></td><td>const conta</td><td><pre>iner = document</pre></td><td>.getElementById('ro</td><td>ot'); 🗕 💻</td><td>LUMU</td><td></td><td></td><td>mpv.</td><td></td><td>9</td><td></td></tr><tr><td></td><td>20</td><td></td><td>const root</td><td>= ReactDOM.crea</td><td><pre>teRoot(container);</pre></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td>21</td><td></td><td><pre>root.render</pre></td><td>(<Button name="</td><td>Save" />);</td><td>_</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td>22</td><td></td><td></td><td></td><td></td><td></td><td>Save</td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td>23</td><td></td><td></script>										
2)	24	<,	/body>										
2	25	<td>tml&gt;</td> <td></td>	tml>										
33													

#### React **JSX**

- JSX is a syntax extension for JavaScript that lets you write HTML-like markup inside a JavaScript file.
- Basically, JSX converts HTML tags into react elements.
- JSX is short for "JavaScript XML".
- JSX looks a lot like HTML, but it is a bit stricter and can display dynamic information.
- JSX and React are two separate things. They're often used together, but you can use them independently of each other. JSX is a syntax extension, while React is a JavaScript library.
- JSX is an extension of the JavaScript language and is translated into regular JavaScript at runtime.
- JSX is used in the Component and Props examples shown earlier.

#### Use Components made by others

- There exists different communities where you can get React Components made by others. Some Examples are:
- Chakra UI: <u>https://chakra-ui.com</u>
- Material UI: <a href="https://mui.com/material-ui/">https://mui.com/material-ui/</a>

### **React in Visual Studio**

Create a new project	[	React     ×     Clear all       All langua ·     All platfor ·     All projec ·
Recent project temp	lates	React App A JavaScript React project template which is bootstrapped by running your
GavaScript Express Application JavaScr	ipt	global install of npx
Windows Forms App	C#	JavaScript Windows macOS Linux Web
ASP.NET Core Web App (Razor Pages)	C#	React App A TypeScript React project template which is bootstrapped by running you global install of npx
■ ASP.NET Core Web API     (native AOT)	C#	TypeScript Windows macOS Linux Web
Console App	C#	React and ASP.NET Core A full-stack application with a frontend React project and a backend ASP.NET Core project
Windows Forms App Visual Ba	isic	JavaScript Windows macOS Linux Web
MSTest Test Project	C#	React and ASP.NET Core A full-stack application with a frontend Peact project and a backend
⊟ Windows Forms App (.NET Framework)	C#	ASP.NET Core project TypeScript Windows macOS Linux Web
		Not finding what you're looking for?

**Different Templates:** 

"React App"

 "React and ASP.NET Core" - Here, the ASP.NET Core project acts as an API backend and the React project acts as the UI.
 You can choose between JavaScript or TypeScript

Back Next

https://learn.microsoft.com/en-us/visualstudio/javascript/create-react-app

https://learn.microsoft.com/en-us/visualstudio/javascript/tutorial-asp-net-core-with-react

### **React App in Visual Studio**

<ul> <li>React App</li> <li>A JavaScript React project template which is bootst global install of npx</li> <li>JavaScript Windows macOS Linux Web</li> </ul>	rapped by running your	<ul> <li>♥ Vite + Reac</li> <li>← C</li> <li>① localhost:53:</li> </ul>	t × +	- □ × A* ☆ ţ= ··· ❹
File Edit View Git Project Build Debug Test Analyze Tools Extensions Wind	low Help <sup>©</sup> Search · reactapp	€ - □ ×		
। • • • • • • • • • • • • • • • • • • •		6 GitHub Copilot 🖻 🖗		
<pre>ain.jsx • × index.html Viscellaneous ·[0"main" ·[*StrictMode -] 14' v import { StrictMode } from 'react' 2 import { createRoot } from 'react-dom/client' 3 import './index.css' 4 import App from './App.jsx' 5 6 createRoot(document.getElementById('root')).render( 7 v (StrictMode) 8   (App /&gt; 9   (/StrictMode), 10  ) 11</pre>	<ul> <li>Solution Explorer</li> <li>Search Solution Explorer (Ctrl+ ")</li> <li>Solution 'reactapp' (1 of 1 project         <ul> <li>Penedencies</li> <li>v.scode</li> <li>public</li> <li>src</li> <li>assets</li> <li>App.css</li> <li>App.jsx</li> <li>index.css</li> <li>gindex.css</li> <li>gindex.css</li> <li>gindex.css</li> <li>gindex.css</li> <li>gindex.css</li> <li>gittingnore</li> <li>Gittingnore</li> <li>Gittingnore</li> <li>Gittingnore</li> <li>Gittingnore</li> <li>Gittingnore</li> <li>README.md</li> <li>picte.config.js</li> <li>index.html</li> <li>pictADME.md</li> <li>vite.config.js</li> </ul> </li> </ul>	ρ]. t)	<b>Vite + React</b> count is 0 Edit src/App.jsx and save to test HMR	
% - • • No issues found Unit 1 Chi 1 SPC 1	GitHub Copilot Chat Solution Explo	rer	Click on the Vite and React logos to learn more	
how output from: Debug	Properties	- # ×		
<pre>VMare: News/Web/silocal/Temp/visualtatic-js-debugger.txt (is) convection. (is) convect</pre>	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1			
eveloper PowerShell Output				
Ready	↑ Add to Source Control •	ect Repository • 🔍 🖉		

#### **React and ASP.NET Core**

#### 👹 React and ASP.NET Core

A full-stack application with a frontend **React** project and a backend ASP.NET Core project

JavaScript Windows macOS Linux Web



Here, the ASP.NET Core project acts as an API backend and the React project acts as the UI.

Solution Explorer Search Solution Explorer (Ctrl+") Solution 'ReactApp1' (2 of 2 projects) reactapp1.client Dependencies vscode public src .gitignore CHANGELOG.md s eslint.config.js lindex.html package.json README.md <sup>15</sup> vite.config.is ReactApp1.Server Connected Services Dependencies Properties appsettings.json CHANGELOG.md Program.cs ReactApp1.Server.http

## **React Visual Studio Resources**

- Create a React app in Visual Studio: <u>https://learn.microsoft.com/en-</u> <u>us/visualstudio/javascript/create-react-app</u>
- Tutorial: Create an ASP.NET Core app with React in Visual Studio: <u>https://learn.microsoft.com/en-us/visualstudio/javascript/tutorial-asp-net-core-with-react</u>
- Create a full stack application by using React and minimal API for ASP.NET Core: <u>https://learn.microsoft.com/en-</u> <u>us/training/modules/build-web-api-minimal-spa/</u>

# Angular



- Developed and maintained by Google.
- Framework for building single-page client applications using HTML and TypeScript.
- Open-source TypeScript framework for client-side applications.
- Angular is component-based architecture.
- Has changed name several times, AngularJS, Angular2, now just known as Angular.
- Basically, the new Angular is using TypeScript, while the older Angular was using JavaScript.
- Homepage: <u>https://angular.dev</u>

### **Angular Playground**

🖌 😫 Playgrou	und • Angular × +				– o x
← → C	25 angular.dev/playground				☆ ② :
(V19 ~) Q	Angular Playground				Select a template Control flow
Ctrl K	main.ts +	23	Ł	Preview	
E Docs <> Tutorials Playground	<pre>1 import {Component} from '@angular/core'; 2 import {bootstrapApplication} from '@angular/platform-browser'; 3 4 @Component({ 5 selector: 'app-root', 6 template: 7 </pre> (h2>Todos 8 <(input #text /> 9 (button (click)="add(text.value)">Add 10 11 @for (todo of todos; track \$index) { 12			Todos Buy Paint Add <sup>2</sup> Buy food <sup>3</sup> Buy Paint	
Ē	14 @if (todo.done) { 15 <<>{{ todo.text }}				
Reference	<pre>10</pre>			Console Terminal Please consider reporting any is angular/angular-cli/issues Initial chunk files   Names main.js   main polyfills.js   polyfills styles.css   styles   Initial to Application bundle generation co Watch mode enabled. Watching for NOTE: Raw file sizes do not refl ations. → Local: http://localhost.4	If you just want to play around with Angular in your browser without setting up a project, you can use the Angular Playground. https://angular.dev/playground
0	<pre>35 } 36 37 bootstranAnnlication(TodosComponent):</pre>			→ press h + enter to show help	P

## Angular in Visual Studio

#### You can also use Visual Studio to create Angular Apps:

Create a new project	Angular × Clear all All langua · All platfor · All projec ·	- 0	×
Recent project templates	🚯 Angular App		
React App JavaScript	A TypeScript Angular project template which is bootstrapped by running your global install of ng and npm		
React and ASP.NET	TypeScript Windows macOS Linux Web		
JavaScript Express Application JavaScript	Angular and ASP.NET Core A full-stack application with a frontend Angular project and a backend ASP.NET Core project		
🖻 Windows Forms App 🛛 C#	TypeScript Windows macOS Linux Web		
ଣ ASP.NET Core Web App C# (Razor Pages)	Not finding what you're looking for?		
■ ASP.NET Core Web API (native AOT) C#	Install more tools and features		
■ Console App C#			
Setup Project			
Windows Forms App Visual Basic			
C#			
□ Windows Forms App (.NET C# Framework)			
	Back	Nex	t

With Visual Studio you can use Angular as frontend and ASP.NET Core As backend

https://learn.microsoft.com/en-us/visualstudio/javascript/tutorial-asp-net-core-with-angular

### Vue

- Vue is another popular JavaScript framework.
- Homepage: <u>https://vuejs.org</u>
- Vue Tutorial:

https://www.w3schools.com/vue/

#### **Vue in Visual Studio**

Create a new project	Vue     ×     Clear all       All langua ·     All platfor ·     All projec ·
Recent project templates	🜐 Vue App
React App JavaScript	A TypeScript Vue project template which is bootstrapped by running your global install of vue and npm
React and ASP.NET Core	TypeScript Windows macOS Linux Web
<sup>©</sup> JavaScript Express Application JavaScript	Wue App A JavaScript Vue project template which is bootstrapped by running your global install of vue and npm
🖻 Windows Forms App 🛛 C#	JavaScript Windows macOS Linux Web
ASP.NET Core Web App C# (Razor Pages)	Vue and ASP.NET Core A full-stack application with a frontend Vue project and a backend ASP.NET Core present.
asp.NET Core Web API (native AOT) C#	JavaScript Windows macOS Linux Web
■ Console App C#	₩ Vue and ASP.NET Core
Setup Project	A full-stack application with a frontend Vue project and a backend ASP.NET
🖻 Windows Forms App Visual Basic	TypeScript Windows macOS Linux Web
MSTest Test Project     C#	
<sup>III</sup> Windows Forms App (.NET Framework) C#	Not finding what you're looking for? Install more tools and features

https://learn.microsoft.com/en-us/visualstudio/javascript/tutorial-asp-net-core-with-vue

# React vs. Angular vs. Vue

- React, Angular and Vue is among the most popular clientside Libraries/Frameworks today.
- They have many similarities but also many differences.
- Angular is developed by Google and uses TypeScript (developed by Microsoft)
- Angular and Vue is Frameworks.
- React is a Front-end UI Library originally created by Facebook.
- All have pros and cons... Which is the best to use? ... I guess there is no easy answer to that question ...

https://www.tatvasoft.com/blog/angular-vs-react-vs-vue/

# **ASP.NET Blazor**

- ASP.NET Blazor is an alternative to miscellaneous frontend JavaScript frameworks.
- ASP.NET Blazor is not a frontend JavaScript framework.
- But it is Microsoft's alternative for a frontend web development framework.
- It uses Razor and C# instead of JavaScript.
- So, if you are already familiar with Visual Studio and C# this is a good alternative.
- And you can use tools from Microsoft for Full stack development.

#### **ASP.NET Blazor in Visual Studio**

			<u></u>				
Create a new project	Blazor × Clear all	- o ×					
Recent project terributers ASP.NET Core Web App (Razor Pages) C# OUNDOWS Forms App C# OUNDOWS Forms App Visual Basic MSTest Test Project OUNDOWS Forms App (.NET OUNDOWS Forms App (.NET) OUNDOWS Forms App (.NET OUNDOWS Forms App (.NET) OUNDOWS FOR Forms App (.NET) OUNDOWS Form (.NET) OUNDOWS Forms Form	<ul> <li>Blazor Web App A project template for creating a Blazor web ap side rendering and client interactivity. This tem apps with rich dynamic user interfaces (UIs).</li> <li>C# Linux macOS Windows Blazor Cloud</li> <li>Blazor Server App A project template for creating a Blazor serve inside an ASP.NET Core app and handles user connection. This template can be used for we interfaces (UIs).</li> <li>C# Linux macOS Windows Blazor Cloud</li> </ul>	op that supports both server-plate can be used for web         Web         ✓       ✓ <td></td> <td></td> <td></td> <td>– ۵ م خ</td> <td>⊇ × @ : \bout</td>				– ۵ م خ	⊇ × @ : \bout
Framework)	<ul> <li>Blazor WebAssembly Standalone App A project template for creating a Blazor app th This template can be used for web apps with i (UIS).</li> <li>C# Linux macOS Windows Blazor Cloud</li> <li>INET MAUI Blazor Hybrid App A project for creating a .NET MAUI application Catalyst, WinUI, and Tizen using Blazor Hybri</li> </ul>	<ul> <li>Home</li> <li>Counter</li> <li>Weather</li> </ul>	Weather This component der Date 02/04/2025	nonstrates showing <b>Temp. (C)</b> 39	g data. <b>Temp. (F)</b> 102	Summary Balmy	
			04/04/2025 05/04/2025 06/04/2025	-10 20 -4 15	4 67 25 58	Cool Balmy Chilly	

#### Hans-Petter Halvorsen

**University of South-Eastern Norway** 

www.usn.no

E-mail: <u>hans.p.halvorsen@usn.no</u> Web: <u>https://www.halvorsen.blog</u>

