

(FED/ID) 1.2

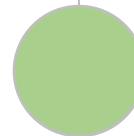
More CSS Media Queries Flex XD



FRONT END DEVELOPMENT / INTERACTIVE DEVELOPMENT

Week 3





FLEXBOX



LABS & GOOD LUCK

Time to delve into code.

Objectives

1. Use the <form></form> tag
2. Use the <input> tag
3. Use the <label></label> tag
4. Write simple validations



The <form> Tag

```
<form action="someurl.php" method="post">  
    |   <!-- All our inputs will go in here -->  
    |</form>
```

action – the URL to send form data to

method – the type of HTTP request

*Until we work with a database noSQL/mongoDB/backend
languages, our forms are actually quite "sad" :(*

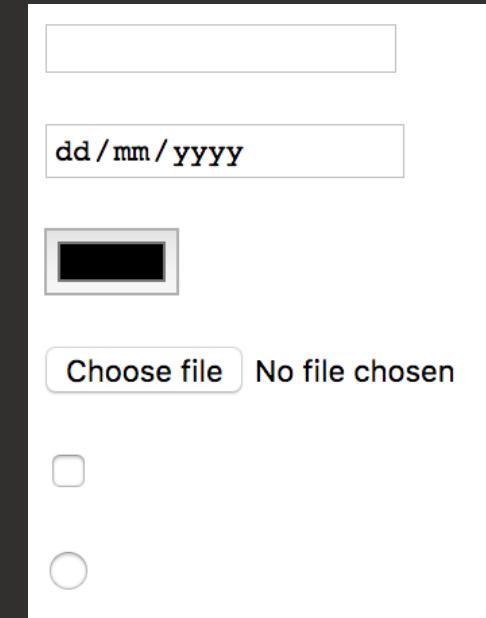
<https://developer.mozilla.org/en-US/docs/Learn/Forms>

The <input> Tag

The input tag creates interactive controls.

The "type" attribute determines the type of input

```
<input type="text">  
  
<input type="date">  
  
<input type="color">  
  
<input type="file">  
  
<input type="checkbox">  
  
<input type="radio">
```



The image shows a white rectangular box containing six examples of input fields, each corresponding to one of the code snippets on the left. From top to bottom: 1. A standard text input field (empty). 2. A date input field showing "dd / mm / yyyy". 3. A color input field showing a solid black square. 4. A file input field with a "Choose file" button and the text "No file chosen". 5. An empty checkbox input field (unchecked). 6. An empty radio input field (unchecked).

A Simple Form

```
<form action="/some-url" method="POST">
  <input type="text" placeholder="hinting text">
  <input type="password" placeholder="Password Pls!">
  <button>Login</button>
  <input type="submit" value="Login Me In">
</form>
```

Let's try sending data.

Sign In

hinting text	Password Pls!	Login	Login Me In
--------------	---------------	-------	-------------

*UX: When designing forms it's a whole process.
Got to think about the platform, device, readability, audience
Also think about are the fields **REALLY NECESSARY?**

Recommending reading:

https://static.lukew.com/webforms_lukew.pdf

Labels

Let's us add captions to the form elements
Labels are really important for making site accessible (visual impairment,

Username: Password:

```
<form action="/some-url" method="GET">
  <label>Username:
    <input type="text" placeholder="hinting text" name="username">
  </label>
  <label>Password:
    <input type="password" placeholder="Password Pls!" name="password">
  </label>

  <button>Login</button>
  <input type="submit" value="Login Me In">
</form>
```

```
<form action="/some-url" method="GET">
  <label for="username">Username:</label>
  <input type="text" placeholder="hinting text" name="username" id="username">
  <label for="password">Password:</label>
  <input type="password" placeholder="Password Pls!" name="password" id="password">
  <button>Login</button>
  <input type="submit" value="Login Me In">
</form>
```

Alternate syntax, using "for" and "id" attributes

Validations

- The 'required' attribute validates that an input is not empty
- There are also type validations. Try changing "type" from "text" to "email"

```
<!-- not every single browser allows form validation -->
<form action="/some-url" method="GET">
    <label for="email">Email:</label>
    <input type="email" placeholder="Email.." name="email" id="email" required>
    <label for="password">Password:</label>
    <input type="password" placeholder="Password Pls!" name="password" id="password" required>
    <button>Login</button>
    <input type="submit" value="Login Me In">
</form>
```

Email: Password:

 Please include an '@' in the email address. 'a' is missing an '@'.

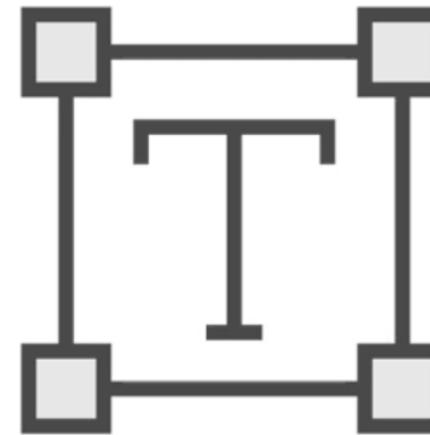
Password:

 Please fill in this field.

Basic Input Categories



Single Line Text



Multiple Line Text



Selection

Basic Single Line Input Fields



Text / Search



Hidden



Password

Additional Input Fields

Allows us to put more semantic meaning to the inputs

<https://repl.it/@malcolmyam/wk03-form-basic>

Email

URL

Telephone

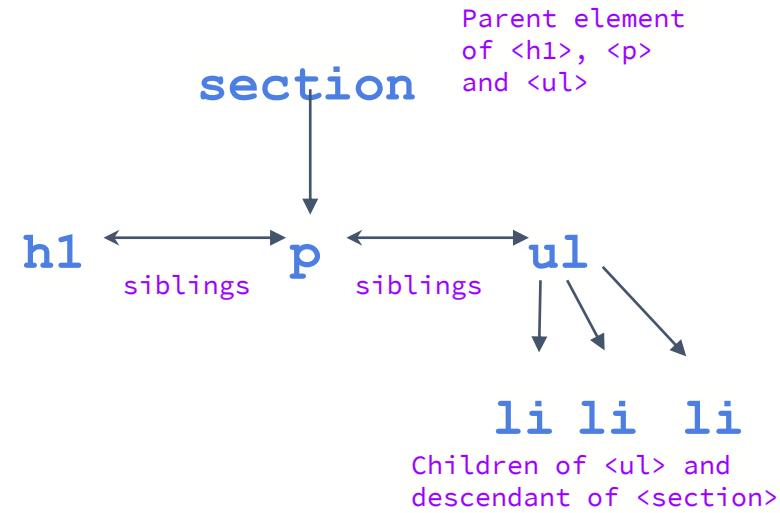
Number

Date

Time

Nested Elements & the DOM

```
<section>
  <h1>Heading</h1>
  <p>Paragraph</p>
  <ul>
    <li>List item</li>
    <li>List item</li>
    <li>List item</li>
  </ul>
</section>
```



Descendant Selector

```
/* Selects only the paragraphs contained within a section*/
```

```
section p{...}
```



Descendant element

```
<section>  
  <p>...</p>  
</section>  
<p>...</p>  
<p>...</p>
```

Descendant Selector

```
/* Selects only links, inside of a paragraph, inside of section */
```

```
section p a { ... }
```

Descendant element

```
<section>
```

```
    <p>There's a <a href="#">link</a> inside this paragraph.</p>
```

```
    <p>Paragraph</p>
```

```
    <a href="#">Link</a>
```

```
</section>
```

```
<a href="#">Link</a>
```

→ Not affected

Descendant Selector

```
.container h1{ }
```

```
.container span{ }
```

```
<section class="container">  
  <h1>Heading</h1>  
  <p>Paragraph with a <span>span</span></p>  
</section>
```

Descendant Selector

You can go as many levels you want with descendant selectors
It is best to use at **most 3 levels**

Grouping Selectors

```
/* applies to all h1 elements */  
h1{...}
```

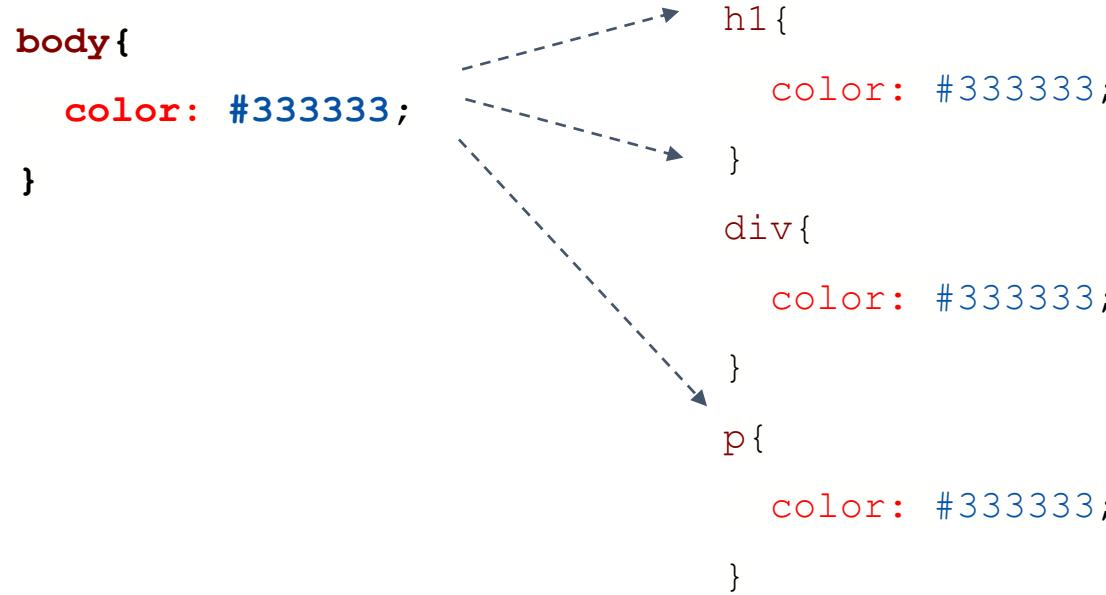
```
/* applies to any element with this class */  
.class{...}
```

```
/* applies to both h1 and h2 elements (grouping) */  
h1, h2{...} /*take note of the spaces when grouping */
```

```
/* applies to all of these elements */  
h1, h2, .class, #id{...}
```

Inheritance

CSS styles can be inherited from the parent elements



All these descendants/child elements **inherit the body style** of color #333333 by default.

<https://repl.it/@malcolmyam/simple-grouping-selectors#index.html>

https://developer.mozilla.org/en-US/docs/Learn/CSS/Building_blocks/Cascade_and_inheritance

Specificity

Specificity determines how browsers decide which CSS rule takes precedence

```
p{color:red; }  
.example{color: blue; }  
  
<p class="example">Red or blue?<p>
```

Specificity

Specificity determines how browsers decide which CSS rule takes precedence

1. Universal (*) <https://repl.it/@malcolmyam/wk03-specificity-demo>
2. type (p)
3. class (.example)
4. id (#example)

Learn more about specificity
<https://specificity.keegan.st/>

Cascade

The cascade in Cascading Style Sheets refer to how style rules are applied based on specificity and source order

Cascade

Style declarations cascade and are read from top to bottom

```
p {  
    font-size: 12px;  
}  
  
/* This style will take precedence*/  
  
p {  
    font-size: 16px;  
}
```

Cascade

```
p {  
    color:black;  
    font-size: 12px;  
    font-size: 16px; /*This style will take precedence */  
}
```

Cascade & Specificity

```
/* This style takes precedence - it has a higher specificity*/  
.example{  
    font-size: 16px;  
}  
  
p{  
    font-size: 12px;  
}
```

Importance !Important

The **!important** keyword overrides source order and specificity.

Added just before the end of the rule declaration before semicolon;

```
p {  
    font-size: 12px !important; /*This style will take precedence */  
}  
  
.example{  
    font-size:16px;  
}
```

Importance !Important

The **!important** keyword overrides source order and specificity.

Added just before the end of the rule declaration before semicolon;

The only way to override **!important** is with another **!important**

This can get messy in the long run.

```
p {  
    font-size: 12px !important;  
}  
  
p {  
    font-size: 18px !important; /*This style will take precedence */  
}
```

Cascading Style Sheets (CSS)

- Defines the **style** of HTML
 - "make all text green"
 - "give the first image a red border"
- The "adjectives" of a webpage



Cascading Style Sheets (CSS)

1. Write properly structured CSS documents
2. Understand various CSS implementations
3. Creating CSS documents



The General Rule

```
selector{  
    property: value;  
    anotherProperty: value;  
}
```

Once again, it's key pair value

key: value

Example

```
/* this is a comment in css. only this is allow */
/* Make all h1's purple and 56px font*/
h1{
    color: □purple;
    font-size: 56px;
}

/* Give all img's a 3px red border */
img{
    border-color: □red;
    border-width: 3px;
}
```

Where do we write our styles?

Inline

```
<h3 style="color: #pink">Let's make me pink</h3>
<p style="color: #black">I'm black. boring.</p>
```

*Use when
"desperate"*

Style Tag

```
<style type="text/css">
  li{
    color: #red;
  }
</style>
```

*Use when you are starting
out*

External Stylesheet

```
<link rel="stylesheet" type="text/css" href="style.css">
```

*Most commonly, best
practice.*

Colors

```
h1{  
    color: purple;  
}  
  
h2{  
    color: cornflowerblue;  
}  
  
h3{  
    color: darkorchid;  
}
```

```
<h1>This is h1</h1>  
<h2>This is confidence</h2>  
<h3>Mini h3 here</h3>
```

This is h1

This is confidence

Mini h3 here

Colors

147 Base Colors for HTML

aliceblue	antiquewhite	aqua	aquamarine	azure	beige
bisque		blanchedalmond	blue	blueviolet	brown
burlywood	cadetblue	chartreuse	chocolate	coral	cornflowerblue
cornsilk	crimson	cyan	darkblue	darkcyan	darkgoldenrod
darkgray	darkgreen	darkkhaki	darkmagenta	darkolivegreen	darkorange
darkorchid	darkred	darksalmon	darkseagreen	darkslateblue	darkslategray
darkturquoise	darkviolet	deeppink	deepskyblue	dimgray	dodgerblue
firebrick	floralwhite	forestgreen	fuchsia	gainsboro	ghostwhite
gold	goldenrod	gray	green	greenyellow	honeydew
hotpink	indianred	indigo	ivory	khaki	lavender
lavenderblush	lawngreen	lemonchiffon	lightblue	lightcoral	lightcyan
lightgoldenrodyellow	lightgray	lightgreen	lightpink	lightsalmon	lightseagreen
lightskyblue	lightslategray	lightsteelblue	lightyellow	lime	limegreen
linen	magenta	maroon	mediumaquamarine	mediumblue	mediumorchid
mediumpurple	mediumseagreen	mediumslateblue	mediumspringgreen	mediumturquoise	mediumvioletred
midnightblue	mintcream	mistyrose	moccasin	navajowhite	navy
oldlace	olive	olivedrab	orange	orangered	orchid
palegoldenrod	palegreen	paleturquoise	palevioletred	papayawhip	peachpuff
peru	pink	plum	powderblue	purple	rebeccapurple
red	rosybrown	royalblue	saddlebrown	salmon	sandybrown
seagreen	seashell	sienna	silver	skyblue	slateblue
slategray	snow	springgreen	steelblue	tan	teal
thistle	tomato	turquoise	violet	wheat	white
	whitesmoke	yellow	yellowgreen		

Colors

Hexadecimal mode

#+ String of 6 hexadecimal numbers (from 0-F)

```
h1{  
    color: #000000;  
}  
  
h2{  
    color: #4b0082;  
}  
  
h3{  
    color: #ff1493;  
}
```

```
<h1>This is h1</h1>  
<h2>This is confidence</h2>  
<h3>Mini h3 here</h3>
```

This is h1

This is confidence

Mini h3 here

Colors

RGB

3 Channels: Red, Green and Blue (Each ranges from 0 – 255)

```
h1{  
    color: □rgb(0,0,0);  
}  
  
h2{  
    color: □rgb(75,0,130);  
}  
  
h3{  
    color: □rgb(255,20,147);  
}
```

```
<h1>This is h1</h1>  
<h2>This is confidence</h2>  
<h3>Mini h3 here</h3>
```

This is h1
This is confidence
Mini h3 here

Colors

RGB

Just like RBG, but with an alpha(transparency) channel. Ranges from 0.0-1.0

```
h1{  
    color: #rgba(11, 99, 150, 1);  
}  
  
h2{  
    color: #rgba(11, 99, 150, 0.6);  
}  
  
h3{  
    color: #rgba(11, 99, 150, 0.2);  
}
```

```
<h1>This is h1</h1>  
<h2>This is confidence</h2>  
<h3>Mini h3 here</h3>
```

This is h1

This is confidence

Mini h3 here

Background

The background property can also set a background

```
body{  
    background: url(https://cdn.pixabay.com/photo/2018/01/05/02/42/background-3062023\_960\_720.jpg);  
}  
  
div{  
    background: rgba(0,0,0,0.7);  
}  
  
p{  
    color: white;  
}  
  
background-repeat: no-repeat;  
background-size: cover;
```

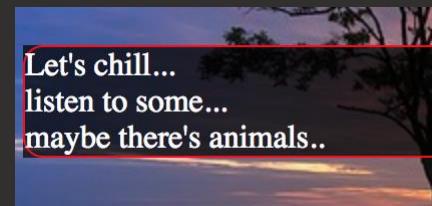
```
<div>  
    <p>Let's chill...<br/>  
        listen to some...<br/>  
        maybe there's animals..</p>
```



Border

Adding some more styles

```
border: 1px solid red;  
border-top: ....  
border-bottom: ...  
border-right: ...  
border-left: ...  
border-radius: 5px
```



ID Selector

Selects an element with a given ID. **Only one per page!**

It is a **best practise** to make your **IDs unique**.

You will know more in Javascript lesson

```
<div>
  <p>You say yes</p>
  <p>You say no</p>
</div>
```

```
<div>
  <p>You say goodbye</p>
  <p id="special">N i Say HellO!</p>
</div>
```

```
div{
  background: □purple;
}
```

```
#special{
  color: □yellow;
}
```

You say yes

You say no

You say goodbye

N i Say HellO!

Class Selector

Selects all elements with a given class

```
<div>
  <p class="highlight">You say yes</p>
  <p>You say no</p>
</div>

<div>
  <p class="highlight">You say goodbye</p>
  <p>N i Say HellO!</p>
</div>
```

```
div{
  background: □purple;
}
.highlight{
  background: □yellow;
}
```

You say yes

You say no

You say goodbye

N i Say HellO!

```
<div id="main">Main section</div>
```

Assign a unique name to a HTML element

An ID, once assigned, shouldn't be repeated

In CSS, an ID is #

```
// change the color of the element with id "main"
#main {
    color: black
}
// change the color of the element with class special
.special {
    color: black
}
```

You can specify a HTML tag and a class to be more specific

// for all <p> with the class special

```
p.special {
```

You can target a children element that is in a parent element

// all <a> inside the ID "main" html element

```
#main a {
```

```
}
```

```
<div class="blue">Main section</div>
<h1 class="blue">Headline</h1>
<h1 class="blue green">Headline</h1>
```

Assign a **name** to a HTML element

Unlike id, classes can be **reused**

Classes can be **stacked**

```
input [type="text"] {color:red;}
```

Styling elements with specific attributes like your forms

ACTIVITY

PLAY GAME TIME
OWN TIME :)

#Practise

<https://flukeout.github.io/>

Instructions

Complete from level 1 to level 11

#Optional challenge level 11 onwards

Read the hints on the sidebar on the page on the right.

The screenshot shows a game interface titled "CSS Diner" with the sub-task "Select the plates". Below the title is a button "Help, I'm stuck!". The main area displays a wooden table with two white plates. In the bottom left corner, there's a CSS editor window with the following code:

```
1 Type in a CSS selector
2 {
3 /* Styles would go here. */
4 }
5
6 /*
7 Type a number to skip to a level.
8 Ex → "5" for level 5
9 */
10
11
12
13
14
15
16
17
18
19
20
```

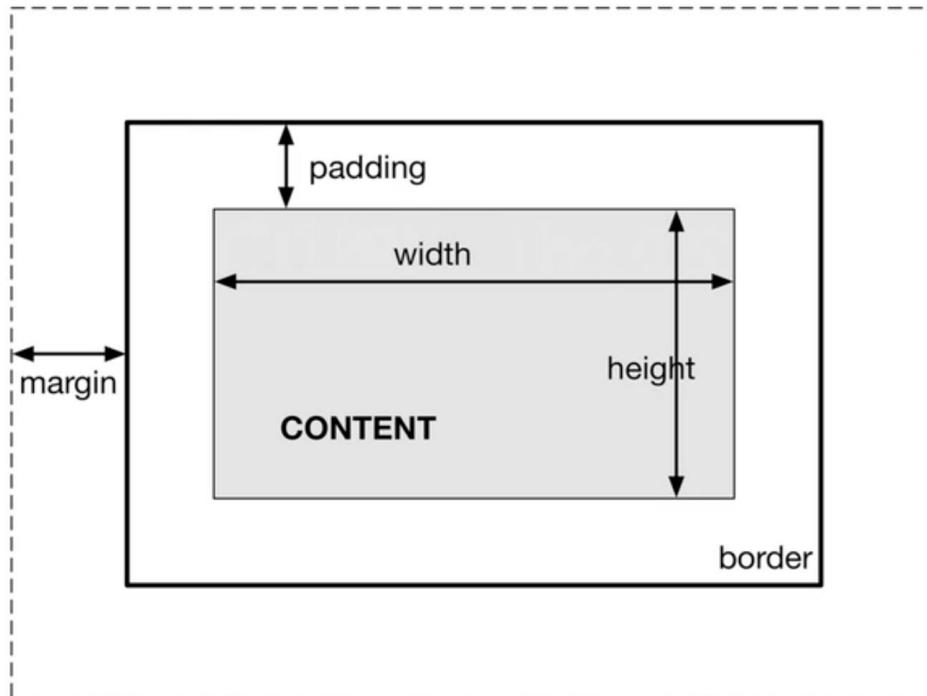
Below the CSS editor is an "HTML Viewer" window showing the following HTML code:

```
<div class="table">
  <plate />
  <plate />
</div>
```

To the right of the game area is a sidebar with the following information:

- Share:** Email, Facebook, Twitter
- Level 1 of 32 ✓**
- Type Selector**: Select elements by their type. A dropdown menu shows "A".

Selects all elements of type A. Type refers to the type of tag, so <div>, <p> and are all different element types.
- Examples**:
 - `div` selects all <div> elements.
 - `p` selects all <p> elements.



CSS Box Model

Content: text, images, etc

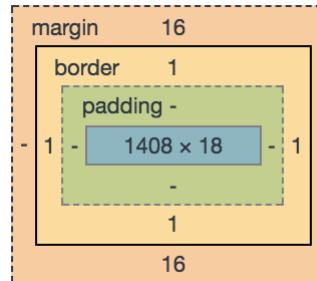
Padding: transparent area around the content, inside of box

Border: goes around the padding & content

Margin: space between boxes

Some paragraph here

```
p {  
    display: block;  
    -webkit-margin-before: 1em;  
    -webkit-margin-after: 1em;  
    -webkit-margin-start: 0px;  
    -webkit-margin-end: 0px;  
}
```



Normalize . A Great Reset script to include
<https://necolas.github.io/normalize.css/>

Great reading source

<https://meyerweb.com/eric/tools/css/reset/>

Default boxes

Several elements have their own default values.

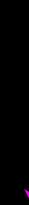
Example: the paragraph shown here has some defaults and could differ from browser to browser.

Hence need to reset them for all browsers to look the same

```
html, body, div, span, applet, object, iframe,  
h1, h2, h3, h4, h5, h6, p, blockquote, pre,  
a, abbr, acronym, address, big, cite, code,  
del, dfn, em, img, ins, kbd, q, s, samp,  
small, strike, strong, sub, sup, tt, var,  
b, u, i, center,  
dl, dt, dd, ol, ul, li,  
fieldset, form, label, legend,  
table, caption, tbody, tfoot, thead, tr, th, td,  
article, aside, canvas, details, embed,  
figure, figcaption, footer, header, hgroup,  
menu, nav, output, ruby, section, summary,  
time, mark, audio, video {  
    margin: 0;  
    padding: 0;  
    border: 0;  
    font-size: 100%;  
    font: inherit;  
    vertical-align: baseline;  
}
```

Include your CSS files in this order

```
<link rel="stylesheet" href="css/normalize.css">  
<link rel="stylesheet" href="css/main.css">
```

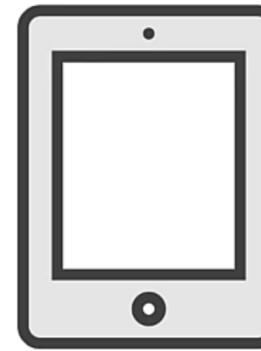


MEDIA QUERIES

Mobile First Design

Media Query

Used to modify the CSS of a site depending on qualities such as viewport size or device type



@media

Applies one or more CSS properties based on the result of a media query

Min-width

If the minimum width of the viewport is at least the specific width or wider, then the media query is true

Max-width

If the maximum width of the viewport is at no more than the specified width, or it's narrower then the media query is true

@media rule

@media type

media feature

```
@media screen and (min-width: 576px)
```

```
{
```

```
/* when the browser viewport is at  
least 576px apply rules here */
```

```
}
```

Mobile First Design

all: matches to all devices

print: matches to print and printer displays

speech: matches to speed reading devices that read out a page

screen: matches all devices that does not match print or speech

Media Features

Media features are used to test a specific feature of the device

```
@media (width: 480px){ }  
@media (orientation: landscape){ }
```

https://developer.mozilla.org/en-US/docs/Web/CSS/Media_Queries/Using_media_queries

```
h1{color: red;}  
/* media query starts here */  
/* Small devices (landscape phones, 576px and up) */  
@media screen and (min-width: 576px) {  
    h1{color: green;}  
}  
/* Medium devices (tablets, 768px and up) */  
@media (min-width: 768px) {  
    h1{color: blue;}  
}  
/* Large devices (desktops, 992px and up) */  
@media (min-width: 992px) {  
    h1{color: purple;}  
}  
/* Extra large devices (large desktops, 1200px and up) */  
@media (min-width: 1200px) {  
    h1{color: lightpink;}  
}
```

MEDIA QUERIES

<https://repl.it/@malcolmyam/simple-media-queries#index.html>

POSITIONING

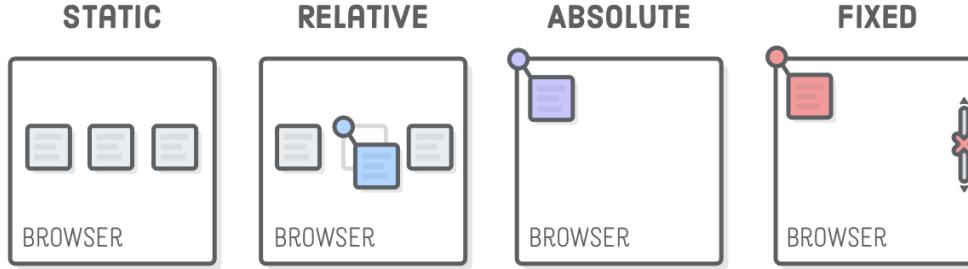
Relative, Absolute,
Fixed

#precisionmatters

Objectives

1. Learn about advanced positioning in CSS
2. Relative, Absolute, Fixed

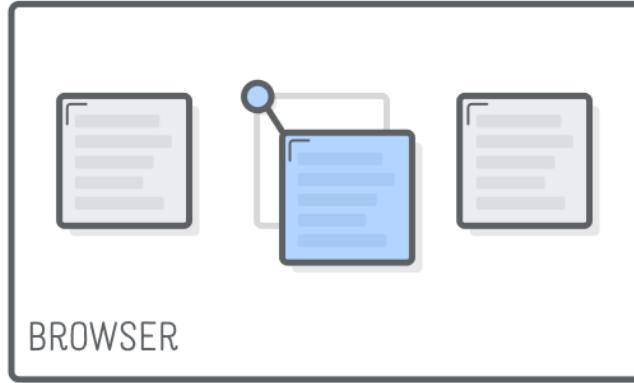




Positioning

CSS position property lets you alter the positioning scheme of a particular element.

Source: <https://internetingishard.com/html-and-css/advanced-positioning/>



RELATIVE POSITIONING

Relative Position

Moves elements around **relative** to where they would appear on the page.

Uses

Nudging boxes around

Source: <https://internetingishard.com/html-and-css/advanced-positioning/>



```
.item-relative {  
  position: relative;  
  bottom: 30px;  
  right: 30px;  
}
```

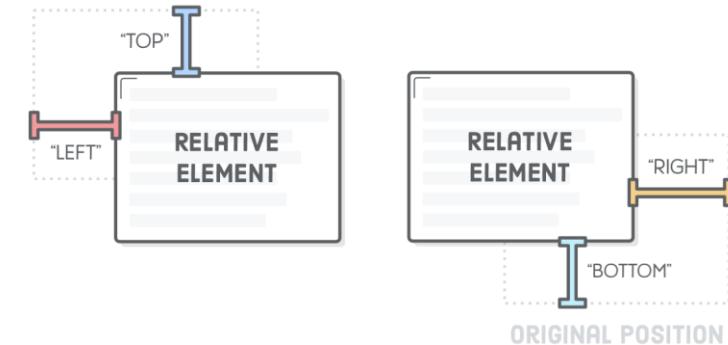
<https://repl.it/@malcolmyam/simple-positioning#index.html>

Relative Position

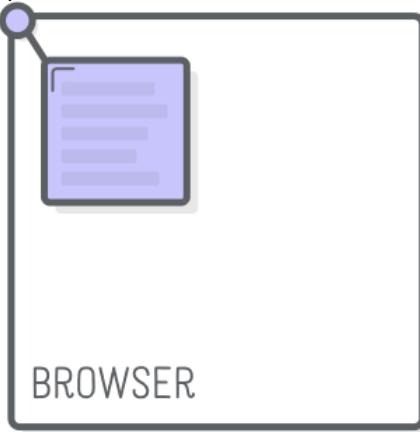
Think of the offsets as being applied after the browser finishes laying out the page.

Uses Top, Left, Right, Bottom properties

ORIGINAL POSITION



Source: <https://internetengineeringishard.com/html-and-css/advanced-positioning/>



ABSOLUTE POSITIONING

- Absolute position elements are placed in relation to a parent content(browser viewport, or parent element).
The parent is the positioning context
- Positions can be modified using positioning offsets, top, right, bottom, left

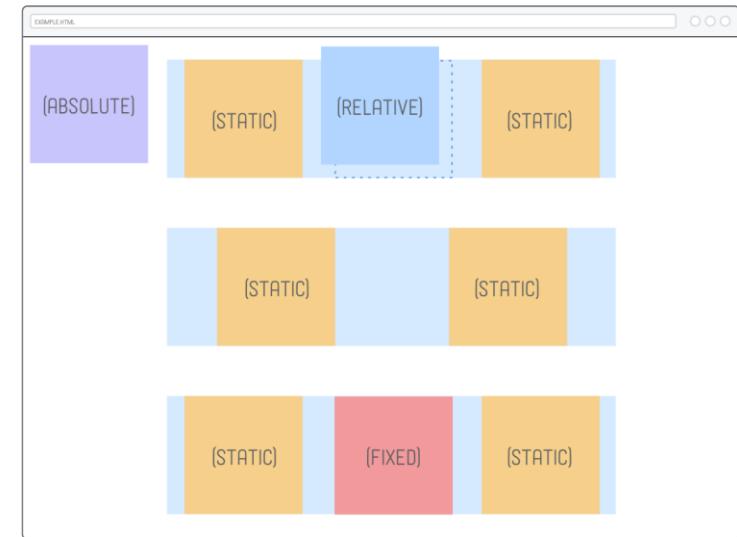
<https://repl.it/@malcolmyam/simple-positioning#index.html>

Reading

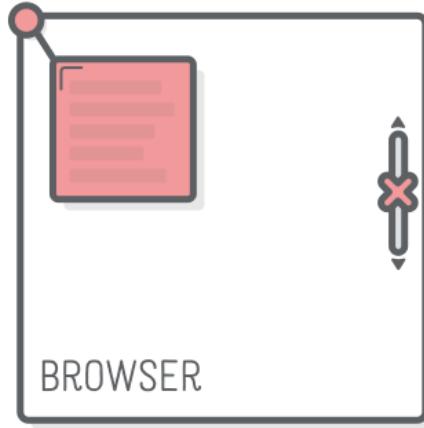
<https://css-tricks.com/almanac/properties/p/position/>

Absolute Position

Offset is relative to the entire browser window instead of the original position of the element.



Source: <https://internettingishard.com/html-and-css/advanced-positioning/>
<https://developer.mozilla.org/en-US/docs/Web/CSS/position>



FIXED POSITIONING

<https://repl.it/@malcolmyam/simple-positioning#index.html>

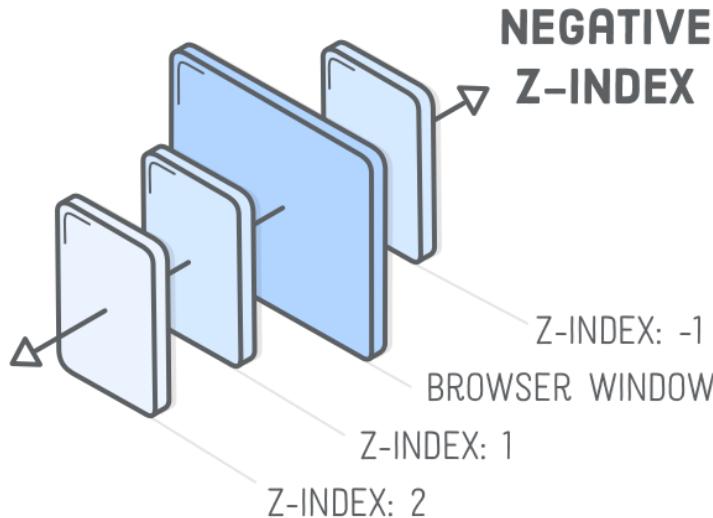
Fixed Position

Very similar to absolute EXCEPT the coordinate system is relative to the entire browser window.

Fixed elements don't scroll with the rest of the page.

Commonly known as sticky elements.

Source: <https://internetingishard.com/html-and-css/advanced-positioning/>



Z-Index

z-index property lets you control the **depth of elements** on the page.

Imagine the screen as a 3D world

Negative z-index go farther into the page. (behind)

Positive z-index comes out of the page

Source: <https://internetingishard.com/html-and-css/advanced-positioning/>

Z-index Reading

<https://developer.mozilla.org/en-US/docs/Web/CSS/z-index>

<https://philipwalton.com/articles/what-no-one-told-you-about-z-index/>

https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Positioning/Understanding_z_index/The_stacking_context

<https://css-tricks.com/almanac/properties/z/z-index/>

[Simple z-index demo](#)

[Z-index animation demo](#)

Z-Index

```
.dropdown > span {
  z-index: 2;
  position: relative; /* This is important! */
  cursor: pointer;
}

.features-menu {
  /* ... */
  z-index: 1;
}
```



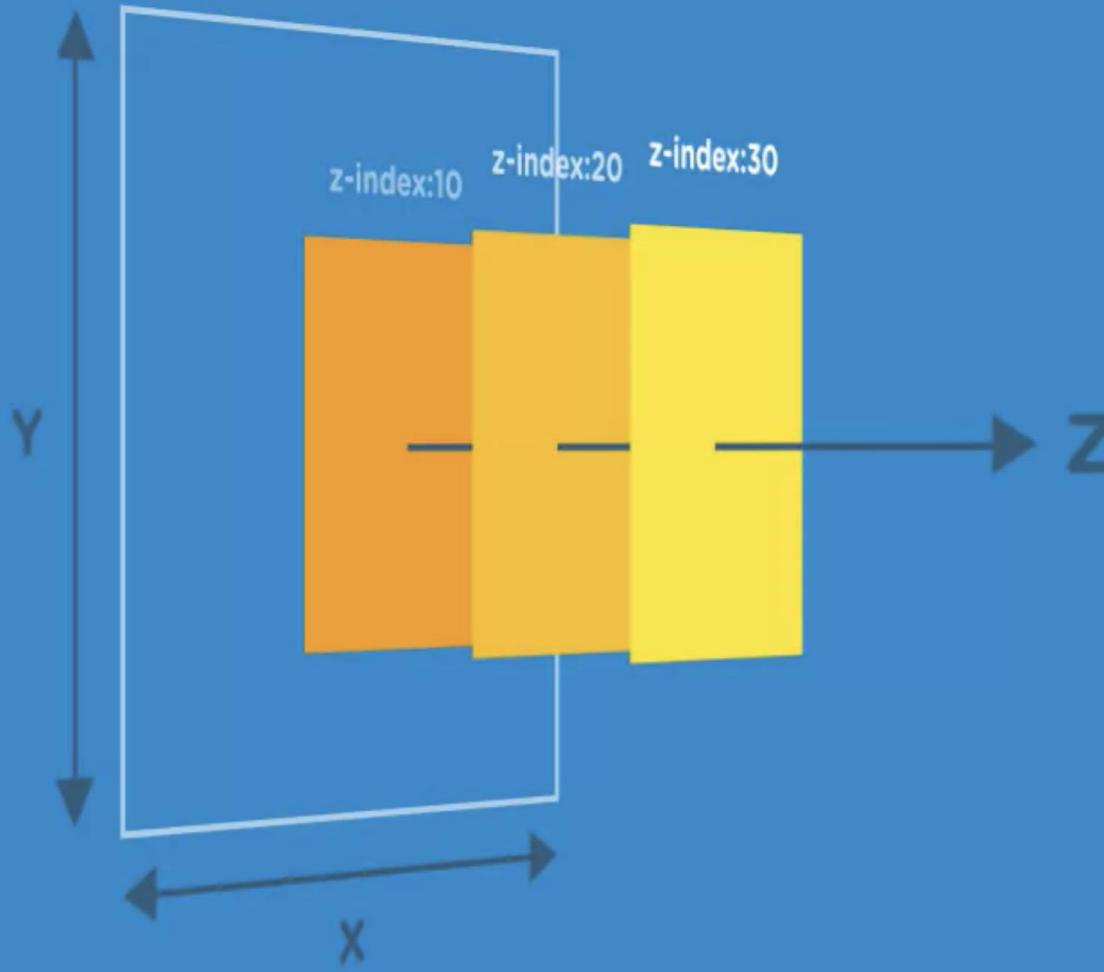
Source: <https://internetingishard.com/html-and-css/advanced-positioning/>

[Simple z-index demo](#)

[Z-index animation demo](#)

Z-index Reading

<https://developer.mozilla.org/en-US/docs/Web/CSS/z-index>
<https://philipwalton.com/articles/what-no-one-told-you-about-z-index/>
https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Positioning/Understanding_z_index/The_stacking_context
<https://css-tricks.com/almanac/properties/z/z-index/>



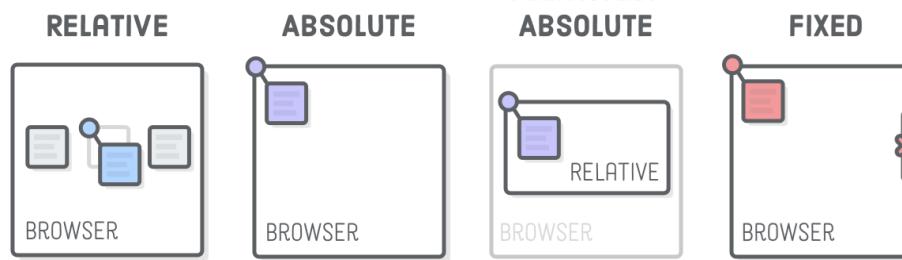
Positioned elements follow a stacking order that determines which elements display above or below other elements.

The z-index property is directly related to stacking order and it's what prevents elements from overlapping other elements.

An element with a higher z-index value overlaps an element with a lower z-index value.

Positioned elements have a z-index of 0 by default.

z-index work ONLY on elements with a position property set to absolute, fixed, or relative. If you set a z-index on an element with no position, it will do nothing.



Positioning

Relative

For tweaking the position of an element without affecting its surrounding boxes

Absolute

Take elements out of the static flow of the page and place relative to the browser window

Relative Absolute

Allows us hook back into the static flow of the page

Fixed

Make elements don't scroll with the rest of the page (sticky)

Source: <https://internetingishard.com/html-and-css/advanced-positioning/>

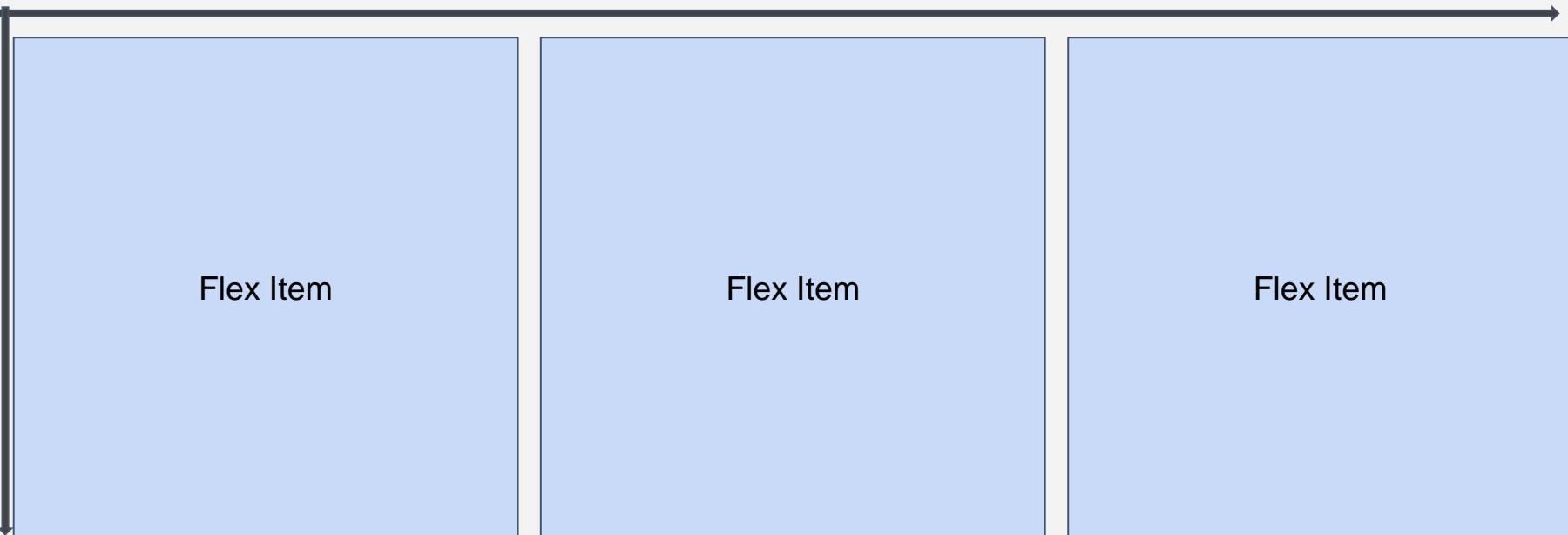
FLEXBOX

FLEXBOX

The flexbox container is the parent of the flex item.

The children of the flexbox container can be laid out in horizontal or vertical easily and flex their size when the content grows or shrink accordingly

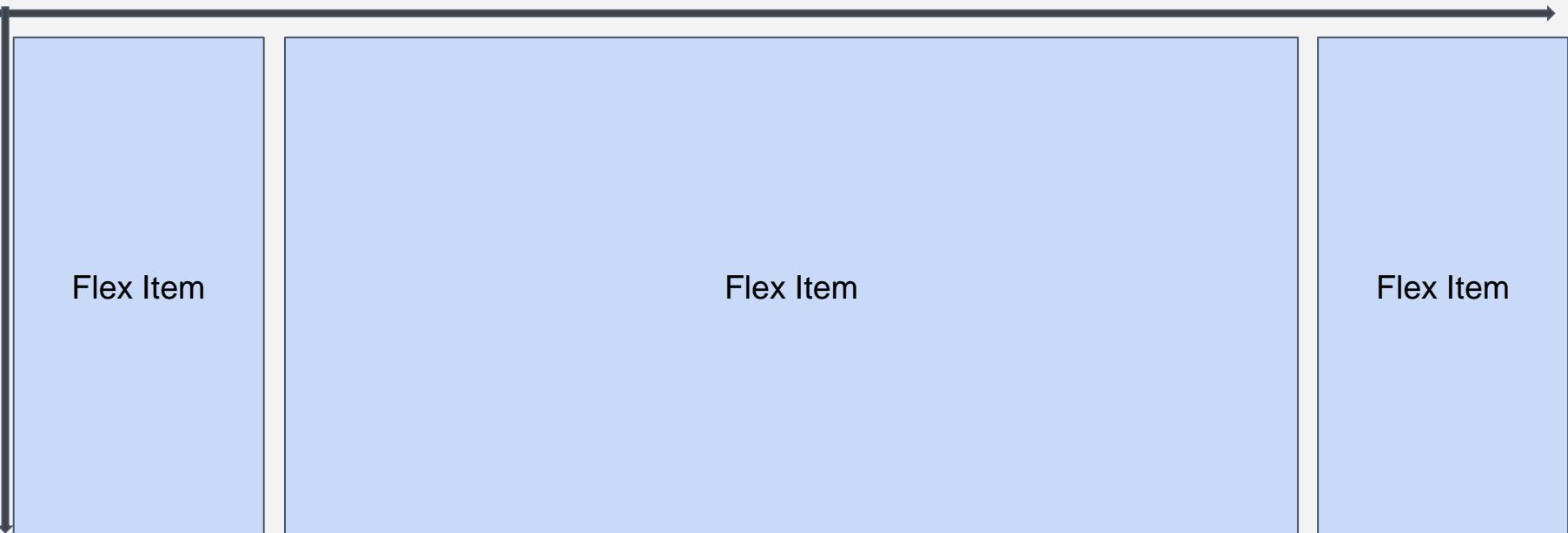
Flexbox Container



FLEXBOX

Apply flexbox to get flexible layouts

Flexbox Container

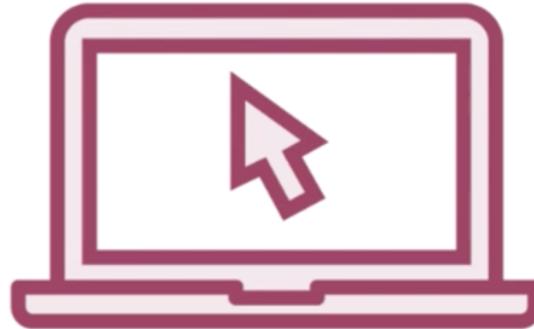


FLEXBOX

Can adapt layout to
suit content



Official (Closed) - Non Sensitive



Breaking down a design

Look at the entire page

See how all main children stack

Start breaking down each row and column

Stop when you can't break it down to stacks



Heading

Sub heading goes here

01/02/20

Bacon ipsum dolor amet brisket porchetta chislic pig boudin cupim jerky beef bresaola andouille alcatra swine, burgdoggen tri-tip spare ribs. Bresaola jerky rump, doner corned beef alcatra fatback chicken capicola salami turke.

Bacon ipsum dolor amet brisket porchetta chislic pig boudin cupim jerky beef bresaola andouille alcatra swine, burgdoggen tri-tip spare ribs. Bresaola jerky rump, doner corned beef alcatra fatback chicken capicola salami turkey frankfurter short ribs. Strip steak hamburger beef, chicken kevin filet mignon ball tip. Prosciutto beef strip steak drumstick hamburger, t-bone kevin sirloin burgdoggen.



Heading

Sub heading goes here



Heading

Sub heading goes here



Heading

Sub heading goes here



Heading

Sub heading goes here

01/02/20

Bacon ipsum dolor amet brisket porchetta chislic pig boudin cupim jerky beef bresaola andouille alcatra swine, burgdoggen tri-tip spare ribs. Bresaola jerky rump, doner corned beef alcatra fatback chicken capicola salami turke.

Bacon ipsum dolor amet brisket porchetta chislic pig boudin cupim jerky beef bresaola andouille alcatra swine, burgdoggen tri-tip spare ribs. Bresaola jerky rump, doner corned beef alcatra fatback chicken capicola salami turkey frankfurter short ribs. Strip steak hamburger beef, chicken kevin filet mignon ball tip. Prosciutto beef strip steak drumstick hamburger, t-bone kevin sirloin burgdoggen.



Heading

Sub heading goes here



Heading

Sub heading goes here



Heading

Sub heading goes here



Heading

Sub heading goes here

01/02/20

Bacon ipsum dolor amet brisket porchetta chislic pig boudin cupim jerky beef bresaola andouille alcatra swine, burgdoggen tri-tip spare ribs. Bresaola jerky rump doner corned beef alcatra fatback chicken capicola salami turke.

Bacon ipsum dolor amet brisket porchetta chislic pig boudin cupim jerky beef bresaola andouille alcatra swine, burgdoggen tri-tip spare ribs. Bresaola jerky rump, doner corned beef alcatra fatback chicken capicola salami turkey frankfurter short ribs. Strip steak hamburger beef, chicken kevin filet mignon ball tip. Prosciutto beef strip steak drumstick hamburger, t-bone kevin sirloin burgdoggen.



Heading

Sub heading goes here



Heading

Sub heading goes here



Heading

Sub heading goes here



Heading

Sub heading goes here

01/02/20

Bacon ipsum dolor amet brisket porchetta chislic pig boudin cupim jerky beef bresaola andouille alcatra swine, burgdoggen tri-tip spare ribs. Bresaola jerky rump, doner corned beef alcatra fatback chicken capicola salami turke.

Bacon ipsum dolor amet brisket porchetta chislic pig boudin cupim jerky beef bresaola andouille alcatra swine, burgdoggen tri-tip spare ribs. Bresaola jerky rump, doner corned beef alcatra fatback chicken capicola salami turkey frankfurter short ribs. Strip steak hamburger beef, chicken kevin filet mignon ball tip. Prosciutto beef strip steak drumstick hamburger, t-bone kevin sirloin burgdoggen.



Heading

Sub heading goes here



Heading

Sub heading goes here



Heading

Sub heading goes here

ACTIVITY

CSS Grid! (20-30 mins)

<https://flexboxfroggy.com/>

Instructions

Complete until level 13

Read the instructions on the page to complete each level

FLEXBOX FROGGY

Welcome to Flexbox Froggy, a game where you help Froggy and friends by writing CSS code! Guide this frog to the lilypad on the right by using the `justify-content` property, which aligns items horizontally and accepts the following values:

- `flex-start`: Items align to the left side of the container.
- `flex-end`: Items align to the right side of the container.
- `center`: Items align at the center of the container.
- `space-between`: Items display with equal spacing between them.
- `space-around`: Items display with equal spacing around them.

For example, `justify-content: flex-end;` will move the frog to the right.

```
1 #pond {  
2   display: flex;  
3 }  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18
```

Flexbox Froggy is created by [CodeSip](#) • [GitHub](#) • [Twitter](#) • [Settings](#)

Want to learn CSS grid? Play [Grid Garden](#).

Next



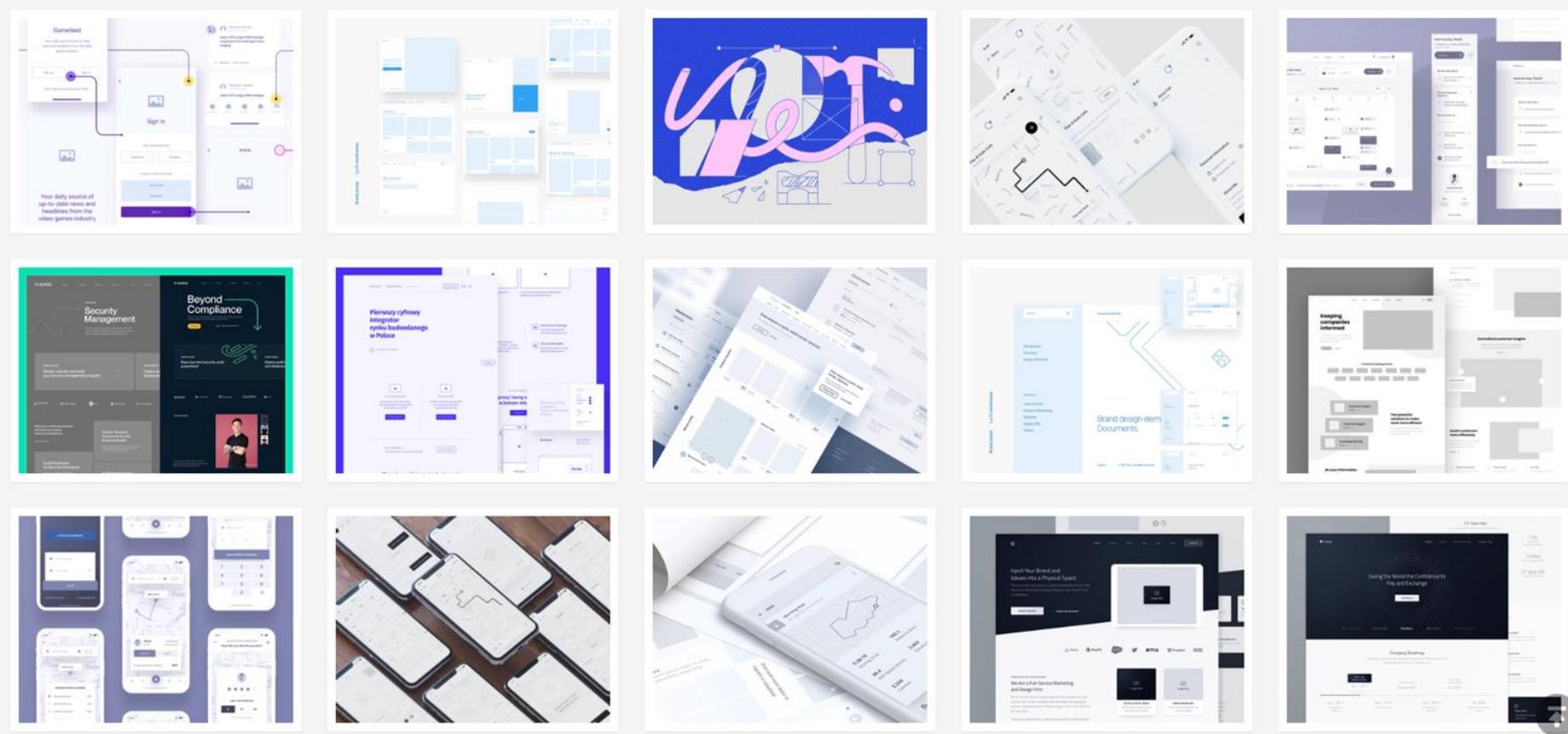
https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Flexible_Box_Layout/Ordering_Flex_Items
<https://webdesign.tutsplus.com/tutorials/a-comprehensive-guide-to-flexbox-ordering-reordering--cms-31564>

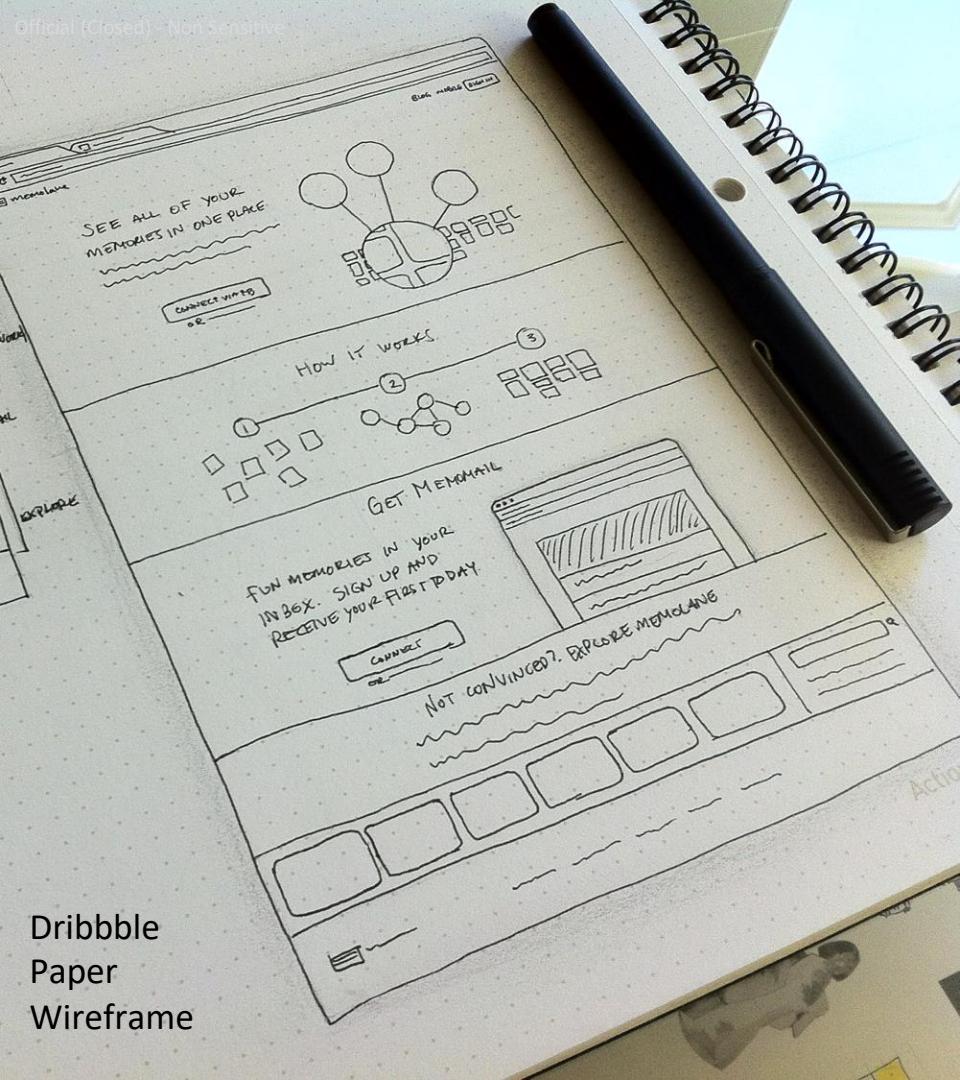
Wireframes

Wireframe

Like a blueprint for a site or app's layout
Shows important functionality
Conveys content types and grouping
Shows information hierarchy

What do wireframes typically look like?





Dribbble
Paper
Wireframe

Online outfit

Boho Holidays

Zwierzęta

Sponsoring

Popularne na sklep

Działki do mówienia

Z Wszystkimi Twoimi zakupy są bezpieczne

Witaj w aplikacji

Zaloguj się

Wyloguj się

Wyślij wiadomość

Invision Wireframe using software

It's also important to understand what wireframes are not.

NOT



Visual Design
Typography
Iconography
Photography

Wireframes are not design.

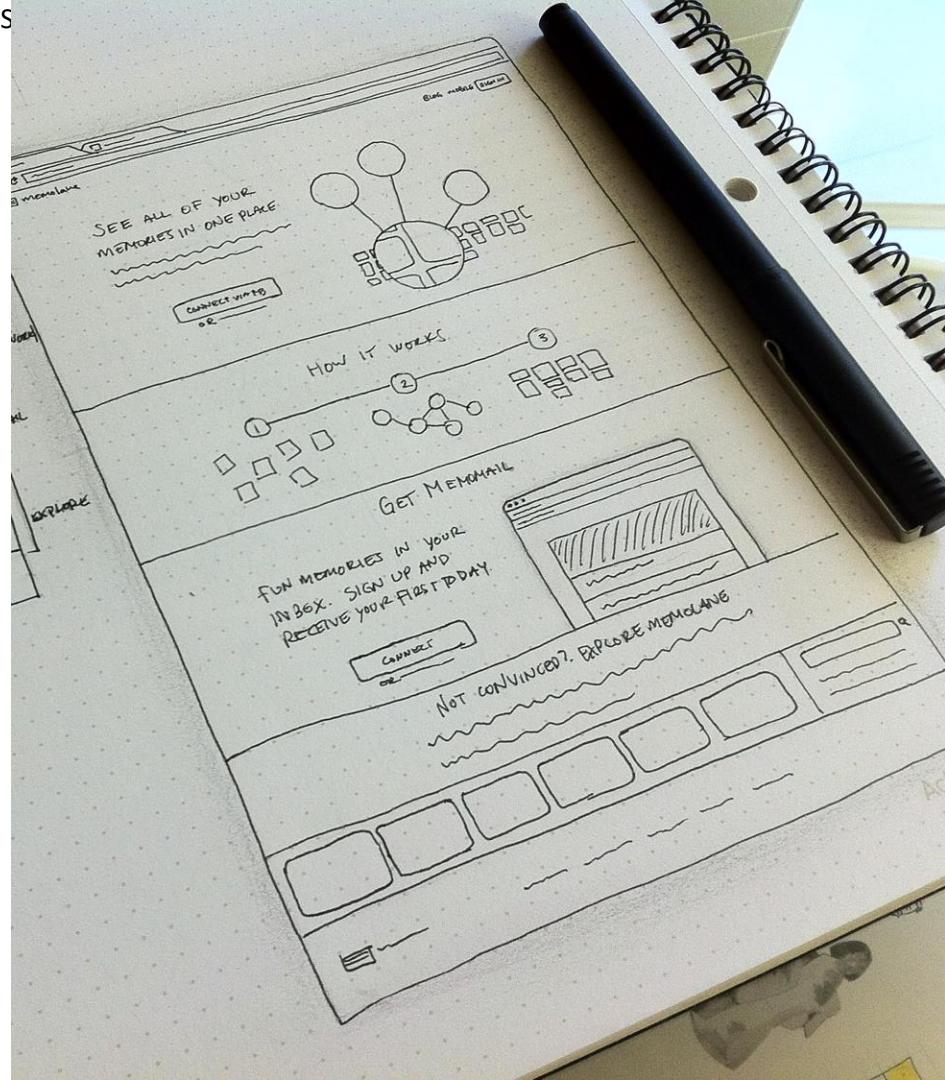
When does wireframing happen



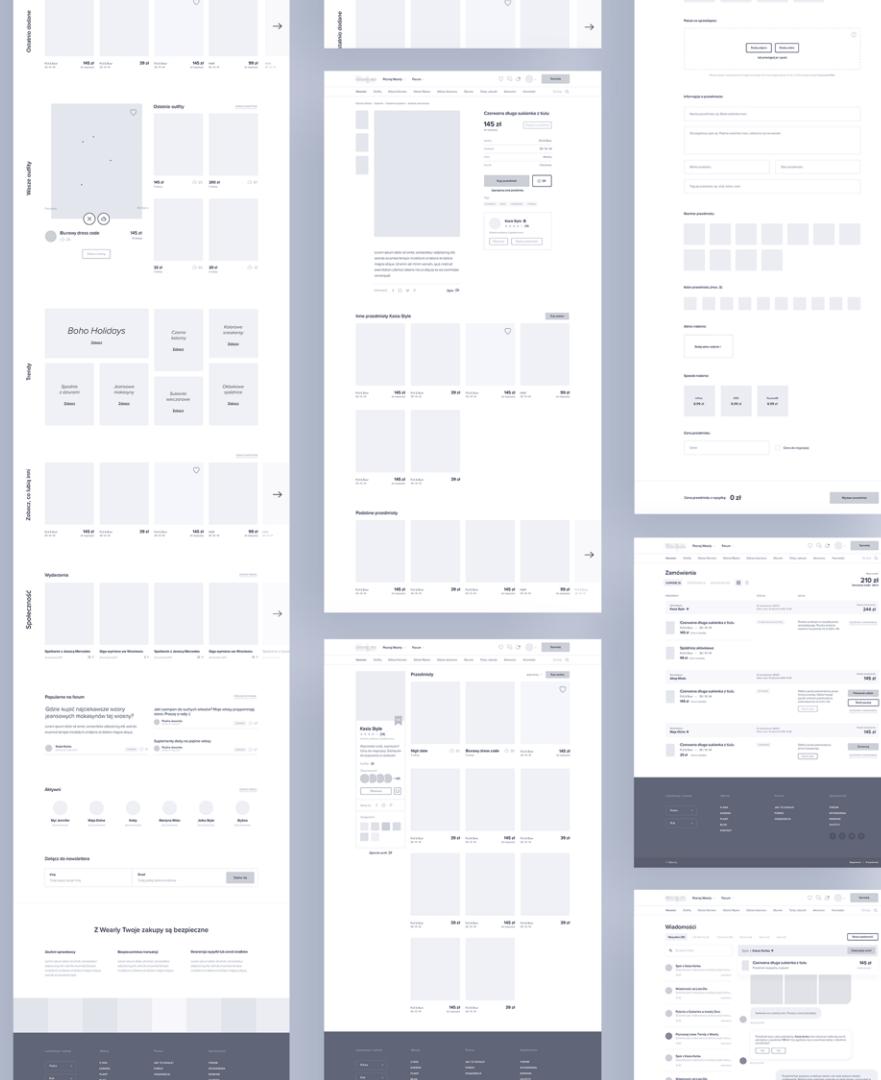
Wireframes can.

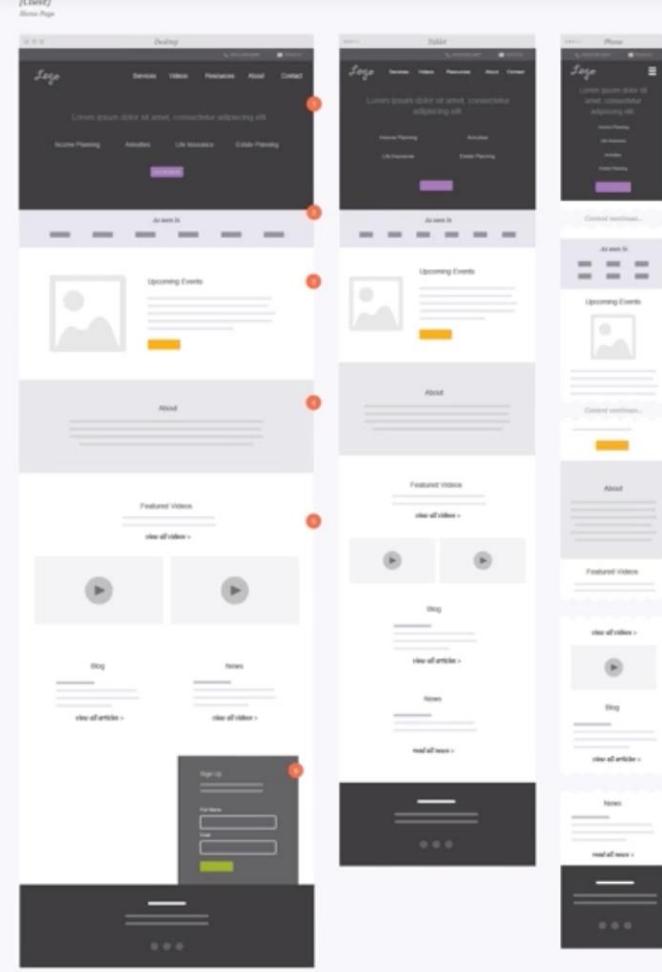
- Quickly generate ideas
- Aids discussion
- Show content hierarchy
- Establish key functionality

Static Wireframe



Static Wireframe





Static Wireframe

Static Wireframe

Pros

Usually quick to create
Variety of tools and software

Cons

Not responsive

Static Wireframing Tools

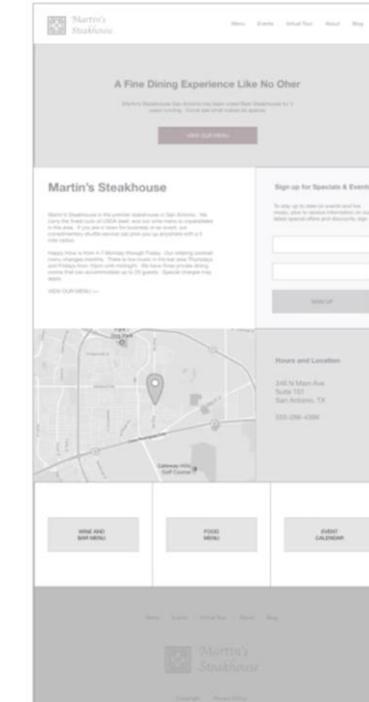
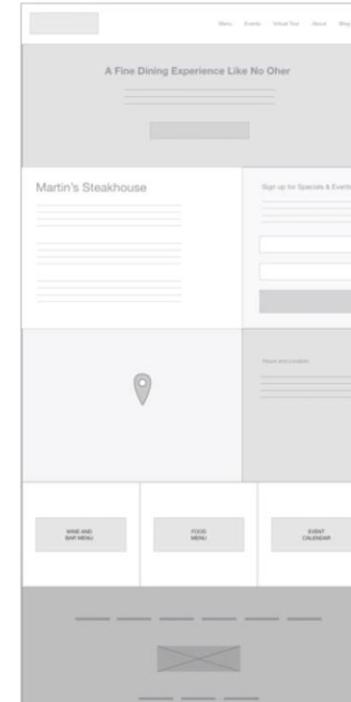
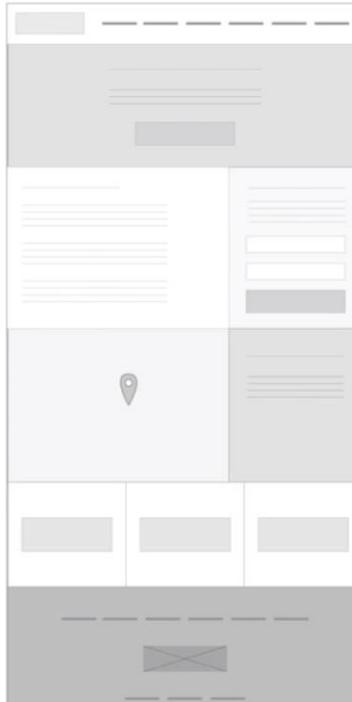
Pen & Paper
Balsamiq
Ommigraffle
InVision
Sketch
Framer
Adobe Illustrator
Adobe Photoshop
Adobe XD
Microsoft Powerpoint

The most important part is not what you use, but how you use it.

Fidelity

The level of detail used to produce a wireframe

Low to High



Low Fidelity Wireframes

Pros

- Can be created quickly
- Get out many ideas before committing to one direction
- Can be simple like a sketch with pen and paper

Cons

- Might not have enough detail for every situation

High Fidelity Wireframes

Pros

Contain more detail than low fidelity
Closer representation of final layout

Cons

Stakeholders might confuse with design



Grid System

A **grid system** is a set of measurements a UI designer can use to align and size objects within the given format. Although there are a few different categories of **grid systems**, there are no strict rules on what can or cannot be a **grid**.



ADOBE XD



<https://www.adobe.com/sea/products/xd.html>

<https://helpx.adobe.com/xd/how-to/what-is-xd.html>

<https://helpx.adobe.com/xd/how-to/web-mobile-app-ui-ux-design-basics.html>

ACTIVITY

CA

#TODO

Complete all CA exercises in the tutorial

ACTIVITY

CA1

#Design and create a simple signup form using [HTML form](#) elements that utilizes form validation.

Your form interface must include AT LEAST the following:

- 3 x text input with different input types (e.g date, time, text, etc)
- 1 x radio button with 4 options
- 1 x checkbox with 4 options
- 1 x text area
- 1 x submit button

Challenge: Apply additional CSS styling for the forms.

Note: For more information refer to tutorial doc.

ACTIVITY

CA2

#Z-index

Hint : <https://developer.mozilla.org/en-US/docs/Web/CSS/position>

Note: For more information refer to tutorial doc.

ACTIVITY

CA3

#Flexboxes

Hint : https://www.w3schools.com/css/css3_flexbox.asp

Note: For more information refer to tutorial doc.

ACTIVITY

CA4 (Optional Challenge)

#Learn how to follow the design and create the layout using html and css.

Note: For more information refer to tutorial doc.



EXTRAS

More practise?

<https://flukeout.github.io/>

General Rule

Tags markup the beginning and the end of an element

<tagName> Some content </tagName>

MDN

<https://developer.mozilla.org/>

Boilerplate

Every HTML document we will start with this boilerplate:

```
<!DOCTYPE html>
<html>
  <head>
    <!-- Our metadata goes here -->
    <title></title>
  </head>
  <body>
    <!-- Our content goes here -->
  </body>
</html>
```

Comments

```
<!-- this is a comment. Doesn't do much-->
```

More HTML elements

```
<p>This is a paragraph</p>
<!-- inline elements & semantics-->
<b>Making text bold</b>
<strong>Making text bold with meaning</strong>
<em>Emphasis on a text</em>
<span>Just an inline element</span>
```

List Items

How about nesting
list?
Inline styles?

```
<!-- ordered list -->
<!--
    |   1. Item 1
    |   2. Item 2
    |   3. Item 3
-->
<ol>
    <li>Item 1</li>
    <li>Item 2</li>
    <li>Item 3</li>
</ol>
```

<https://developer.mozilla.org/en-US/docs/Web/HTML/Element/ol>

```
<!-- unordered list-->
<!--
    |   - Item 1
    |   - Item 2
    |   - Item 3
-->
<ul>
    <li><b>Item</b>1</li>
    <li>Item 2</li>
    <li>Item 3</li>
</ul>
```

<https://developer.mozilla.org/en-US/docs/Web/HTML/Element/ul>

div

The **HTML Content Division element (<div>)** is the generic container for flow content. It has no effect on the content or layout until styled using CSS.

As a "pure" container, the <div> element does not inherently represent anything.

```
<!--acts like a generic container-->
<div>
    Lorem ipsum.. Insert some content
    <p>or maybe a paragraph</p>
    <ul>
        <li>just item 1</li>
        <li>just item 2</li>
    </ul>
</div>
```

*Demo inspector
mode*

< a > </ a > Hyperlink

href attribute

Works link an anchor

Can be an internal (#bookmark)

Can be an external link

```
<a href="https://www.google.com"  
target="_blank">Google</a>
```

```
target="_blank" //open link in new window or tab  
target="_self" //open link in same frame  
target="_parent" //open link in parent frame  
target="_top" //open link in full body of window
```

 Image

alt attribute

allows alternate text to display when image is not available

title attribute (hyperlinks can use this too!)

Displays a string of text upon mouse hover

```

```

Attributes

Adding additional Information to Tags

<tag name="value"></tag>

```
<div>
  

  <h1 class="highlight">John Doe</h1>

  <a href="www.google.com">Gooooogle</a>

  <link rel="stylesheet" type="text/css" href="style.css">
</div>
```

<https://developer.mozilla.org/en-US/docs/Web/HTML/Attributes>

Images

Adding some "pretty" at last.

```

```

Self-closing tag



Demo

Relative source
Absolute
source

LEARNING POINTS

- Name your files using **lower-case** (in web dev, files are hosted on the server. Having About.html about.html are treated differently by the server.)
- Name your **files with intent and meaning** (e.g Contact Us page should be contact-us.html and not page3.html)
- Have spaces in your file names? Replace them with dashes "-" instead. This reduces conflict when people are typing and it's also easier to recognise.
- In web development, **ALWAYS ensure that you have the proper file extension**. For Windows users, it is highly recommended you enable file extensions in your system.
- Submission criteria -> Create a zip file of all your work. (services.html.. Home.html -> combine to form a zip file)
- Students mindset? -> Simply renaming the file report.docx to report.pdf. Will it open? NO!

Simple reason. You see an existing HTML file in the file browser, it is called services. Then you decide to give it an extension services.html. You have file extensions hidden. The full file name will be called services.html.html.

“Awesome” file naming..
IDWk01-Pentatonix.**pptx.pptx**
IDWk01_TeamYoutube.**pptx.pptx**

<https://www.howtogeek.com/205086/beginner-how-to-make-windows-show-file-extensions/>

<https://support.winzip.com/hc/en-us/articles/115011457948-How-to-configure-Windows-to-show-file-extensions-and-hidden-files>

