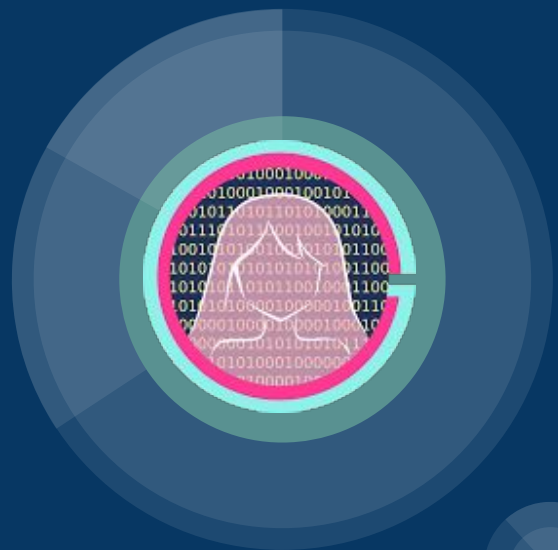


# CSS Lecture 2

Code Girls 2020-21



The <div> element is a block-level element.



## Display

- The **display** property specifies if/how an element is displayed.
- Every HTML element has a default **display** value
- Usually block or inline
- Block: starts on a new line and takes up the full width available
  - <div>
  - <h1> - <h6>
  - <p>
  - <form>
  - <header>
  - <footer>
  - <section>
- Inline: does not start on a new line and only takes up as much width as necessary
  - <span>
  - <a>
  - <img>
- None: hide and show elements without deleting and recreating them
  - <script>

This is an inline <span> element inside a paragraph.



# display: inline-block

- **display: inline-block** is different from **display: inline** because
  - It allows you to set a width and height on the element
  - The top and bottom margins/paddings are respected
- **display: inline-block** is different from **display: block** because
  - It does not add a line-break after the element and can sit next to other elements
- Examples:
  - [https://www.w3schools.com/css/tryit.asp?filename=trycss\\_inline-block\\_spn1](https://www.w3schools.com/css/tryit.asp?filename=trycss_inline-block_spn1)
- Creating Navigation Links:
  - [https://www.w3schools.com/css/tryit.asp?filename=trycss\\_inline-block\\_nav](https://www.w3schools.com/css/tryit.asp?filename=trycss_inline-block_nav)



## Width and Max-width

- Set the width of a block-level element to prevent it from stretching out to the edges of its container
- You can set the margins to auto, to horizontally center the element within its container.
- Using width will cause the text to go off the screen if your screen is too small so if you are making a website for a smaller screen, use max-width

```
div.ex1 {  
  width: 500px;  
  margin: auto;  
  border: 3px solid #73AD21;  
}
```

```
div.ex2 {  
  max-width: 500px;  
  margin: auto;  
  border: 3px solid #73AD21;  
}
```



# Position

```
div.sticky {  
  position: -webkit-sticky; /* Safari */  
  position: sticky;  
  top: 0;  
  background-color: green;  
  border: 2px solid #4CAF50;  
}
```

```
div.absolute {  
  position: absolute;  
  top: 80px;  
  right: 0;  
  width: 200px;  
  height: 100px;  
  border: 3px solid #73AD21;  
}
```

```
div.static {  
  position: static;  
  border: 3px solid #73AD21;  
}
```

```
div.relative {  
  position: relative;  
  left: 30px;  
  border: 3px solid #73AD21;  
}
```

```
div.fixed {  
  position: fixed;  
  bottom: 0;  
  right: 0;  
  width: 300px;  
  border: 3px solid #73AD21;  
}
```

```
img {  
  position: absolute;  
  left: 0px;  
  top: 0px;  
  z-index: -1;  
}
```

- The position property specifies the type of positioning method used for an element.
- Use the top, right, bottom, and left properties to adjust the positions
- **static**
  - The default
- **relative**
  - Positioned relative to its normal position
- **fixed**
  - Positioned relative to the viewport (the visible area of your webpage)
  - Always stays in the same place even if the page is scrolled
  - It does not leave a space in the place where it would have been located and goes over the text
- **absolute**
  - Positioned relative to the nearest positioned ancestor (anything except static)
  - If it has no positioned ancestors, it uses the document body, and moves along with page scrolling
- **sticky**
  - toggles between **relative** and **fixed**, depending on the scroll position
  - Positioned relative until you scroll to a certain point and then it sticks in place
  - This may only work on certain web browsers
- Elements may be overlapped
  - Use the **z-index** property to specify if the element should be placed in front or behind other elements



# Overflow

- The **overflow** property controls what happens when the content is too large to fit the area
- Only works for block elements with a specified height
- **visible**
  - The default setting
  - The overflow is not clipped and the content goes outside the box
- **hidden**
  - The overflow is clipped and the content that's outside the box is hidden
- **scroll**
  - The overflow is clipped and a scrollbar is added to see the rest of the content (both vertically and horizontally)
- **auto**
  - Similar to scroll, but it adds scrollbars only when necessary

You can use the overflow property when you want to have better control of the layout. The overflow property specifies what happens if content overflows an element's box.

You can use the overflow property when you want to have better control of the layout. The overflow

You can use the overflow property when you want to have better control of

You can use the overflow property when you want to have better control of



# Float

- Used for positioning and formatting content
  - For example, let an image float left to the text in a container
- left
- right
- none
  - Default
  - The element does not float (will be displayed just where it occurs in the text)
- inherit
  - The element inherits the float value of its parent

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus imperdiet, nulla et dictum interdum, nisi lorem egestas odio, vitae scelerisque enim ligula venenatis dolor. Maecenas nisl est, ultrices nec congue eget, auctor vitae massa. Fusce luctus vestibulum augue ut aliquet. Mauris ante ligula, facilisis sed ornare eu, lobortis in odio. Praesent convallis urna a lacus interdum ut hendrerit risus congue. Nunc sagittis dictum nisi, sed ullamcorper ipsum dignissim ac...



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus imperdiet, nulla et

dictum interdum, nisi lorem egestas odio, vitae scelerisque enim ligula venenatis dolor. Maecenas nisl est, ultrices nec congue eget, auctor vitae massa. Fusce luctus vestibulum augue ut aliquet. Mauris ante ligula, facilisis sed ornare eu, lobortis in odio. Praesent convallis urna a lacus interdum ut hendrerit risus congue. Nunc sagittis dictum nisi, sed ullamcorper ipsum dignissim ac...



# Clear and Clearfix

- The **clear** property specifies what elements can float beside the cleared element and on which side.
- Usually used after you've used the float element
  - The clear should be the same as the float
    - For example, if you floated to the left, you should clear to the left
    - Your floated element will continue to float, but the cleared element will appear below it on the web page.
- **none**
  - Default
  - Allows floating elements on both sides
- **left**
  - No floating elements allowed on the left side
- **right**
  - No floating elements allowed on the right side
- **both**
  - No floating elements allowed on either the left or the right side
- **inherit**
  - The element inherits the clear value of its parent
- **clearfix**
  - If an element is taller than the element containing it, and it is floated, it will "overflow" outside of its container:



## Without clear

div1

div2 - Notice that div2 is after div1 in the HTML code. However, since div1 floats to the left, the text in div2 flows around div1.

## With clear

div3

div4 - Here, `clear: left;` moves div4 down below the floating div3. The value "left" clears elements floated to the left. You can also clear "right" and "both".

```
.div1 {
  float: left;
  width: 100px;
  height: 50px;
  margin: 10px;
  border: 3px solid #73AD21;
}

.div2 {
  border: 1px solid red;
}

.div3 {
  float: left;
  width: 100px;
  height: 50px;
  margin: 10px;
  border: 3px solid #73AD21;
}

.div4 {
  border: 1px solid red;
  clear: left;
}
```

## Without Clearfix

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus imperdiet, nulla et dictum interdum...



## With Clearfix

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus imperdiet, nulla et dictum interdum...



```
.clearfix {
  overflow: auto;
}
```



# Combinators

- A CSS selector can contain more than one simple selector and between the simple selectors, we can include a combinator.
  - [https://www.w3schools.com/css/css\\_combinators.asp](https://www.w3schools.com/css/css_combinators.asp)



# Pseudo-class

- Used to define a special state of an element
- Examples of what we can do with a pseudo-class
  - Style an element when a user mouses over it
  - Style visited and unvisited links differently
  - Style an element when it gets focus
- Note:
  - **a:hover** MUST come after **a:link** and **a:visited**
  - **a:active** MUST come after **a:hover**
- [https://www.w3schools.com/css/tryit.asp?filename=trycss\\_pseudo-class](https://www.w3schools.com/css/tryit.asp?filename=trycss_pseudo-class)
- [https://www.w3schools.com/css/tryit.asp?filename=trycss\\_pseudo-class\\_hover\\_div](https://www.w3schools.com/css/tryit.asp?filename=trycss_pseudo-class_hover_div)
- [https://www.w3schools.com/css/tryit.asp?filename=trycss\\_pseudo-class\\_hover\\_tooltip](https://www.w3schools.com/css/tryit.asp?filename=trycss_pseudo-class_hover_tooltip)
- The **:first-child** pseudo-class matches a specified element that is the first child of another element
  - [https://www.w3schools.com/css/tryit.asp?filename=trycss\\_first-child1](https://www.w3schools.com/css/tryit.asp?filename=trycss_first-child1)
- Match the first <i> element in all <p> elements:
  - [https://www.w3schools.com/css/tryit.asp?filename=trycss\\_first-child2](https://www.w3schools.com/css/tryit.asp?filename=trycss_first-child2)

```
/* unvisited link */  
a:link {  
    color: #FF0000;  
}
```

```
/* visited link */  
a:visited {  
    color: #00FF00;  
}
```

```
/* mouse over link */  
a:hover {  
    color: #FF00FF;  
}
```

```
/* selected link */  
a:active {  
    color: #0000FF;  
}
```



# Pseudo-elements

- Used to style specified parts of an element
- Examples:
- **::first-line** adds a special style to the first line of a text
  - [https://www.w3schools.com/css/tryit.asp?filename=trycss\\_firstline](https://www.w3schools.com/css/tryit.asp?filename=trycss_firstline)
- **::first-letter** adds a special style to the first letter of a text
  - [https://www.w3schools.com/css/tryit.asp?filename=trycss\\_firstletter](https://www.w3schools.com/css/tryit.asp?filename=trycss_firstletter)
- Pseudo-elements can also be combined with CSS classes
  - [https://www.w3schools.com/css/tryit.asp?filename=trycss\\_pseudo-element](https://www.w3schools.com/css/tryit.asp?filename=trycss_pseudo-element)
- You can also combine several pseudo-elements
  - [https://www.w3schools.com/css/tryit.asp?filename=trycss\\_firstline\\_letter](https://www.w3schools.com/css/tryit.asp?filename=trycss_firstline_letter)
- **::before** and **::after** are used to insert content before or after the content of an element
  - [https://www.w3schools.com/css/tryit.asp?filename=trycss\\_before](https://www.w3schools.com/css/tryit.asp?filename=trycss_before)
  - [https://www.w3schools.com/css/tryit.asp?filename=trycss\\_after](https://www.w3schools.com/css/tryit.asp?filename=trycss_after)
- **::selection** matches the portion of an element that is selected by a user
  - Can be applied to **color**, **background**, **cursor**, and **outline**
  - [https://www.w3schools.com/css/tryit.asp?filename=trycss3\\_selection](https://www.w3schools.com/css/tryit.asp?filename=trycss3_selection)



# Opacity/Transparency

- Opacity
  - A value from 0.0 to 1.0
  - Low values are more transparent and higher values are less transparent
  - The default opacity is 1
- The **opacity** property can be used with the **:hover** selector to change the opacity when your mouse hovers over the image
  - [https://www.w3schools.com/css/tryit.asp?filename=trycss\\_image\\_transparency](https://www.w3schools.com/css/tryit.asp?filename=trycss_image_transparency)
- If you use the opacity property to add transparency to the background of an element, all of its child elements inherit the same transparency, which may make the text hard to read
  - If you do not want this, use RGBA color values to specify the opacity of the background
- Text in transparent box
  - [https://www.w3schools.com/css/tryit.asp?filename=trycss\\_transparency](https://www.w3schools.com/css/tryit.asp?filename=trycss_transparency)

```
img {  
  opacity: 0.5;  
}
```

```
img:hover {  
  opacity: 1.0;  
}
```