



# Beginning HTML and CSS

## Class 4

# Welcome!

Girl Develop It is here to provide affordable and accessible programs to learn software through mentorship and hands-on instruction.

## Some "rules"

- We are here for you!
- Every question is important
- Help each other
- Have fun

# Static Positioning

HTML elements are positioned static by default.

Static elements are positioned in the normal flow of the page

Static elements ignore top, bottom, right or left property specifications.

# Static Positioning

In normal flow, inline boxes flow from left to right, wrapping to next line when needed.

```
  
  

```



# Static Positioning

In normal flow, block boxes flow from top to bottom, making a new line after every box.

```
<p>Greetings</p>  
<p>Hello</p>  
<p>Hi there!</p>
```

Greetings

Hello

Hi there!

# Relative Positioning

Makes an element a "positioning context" in which to position other elements relative to it.

Relative positioning and absolute positioning are used together.

Use the properties top, bottom, right and left to relative position an element.

# Relative Positioning

The "relative" value will still put the element in the normal flow, but then offset it according to top/left/right/bottom properties.

```
.relative{  
  position: relative;  
  left: 80px;  
  top: 20px;  
  height: 100px;  
  background-color: yellow;  
}
```

Hello, hi!

# Absolute Positioning

Determined by its offset values in the properties top, bottom, right and left.

An absolutely positioned element is offset from its container block, positioned relative.

Its container block is the first element that has a position other than static.

If no such element is found, the container block is <html>.



The "absolute" value will take the element out of the normal flow and position it in relation to the window (or the closest non-static element).

```
.top{  
  position: absolute;  
  top: 10px;  
  right: 10px;  
  background-color: yellow  
}
```

```
.top{  
  position: absolute;  
  bottom: 10px;  
  left: 60px;  
  background-color: green  
}
```

The z-index property specifies the stack order of positioned elements, in the case that they are overlapping. The element with highest z-index goes on top.

```
.bottom{  
  position: absolute;  
  bottom: 10px;  
  left: 60px;  
  background-color: yellow;  
}
```

```
.top{  
  position: absolute;  
  bottom: 15px;  
  left: 60px;  
  background-color: green;  
  z-index: 2;  
}
```

# Float

An element can be pushed to the right or the left, allowing others to wrap around it.

Text and other content will flow around a floated element.

You can specify a) whether an element is floated or not, and b) which side it floats on.

# Float

```
.float{  
  float:left;  
  width:50px;  
  background:yellow;  
}
```

hi Not floating element  
hi Not floating element  
hi

# Clear

Specifies which side of an element floats are not allowed.

Clearing both sides makes sure floats don't flow past the clear element.

```
clear: right;  
clear: left;  
clear: both;
```

# Clear

```
.float{  
  float:left;  
  width:50px;  
  background:yellow;  
}  
.clear{  
  clear:both  
}
```

hi Not floating element

hi

hi

Not floating element

# Let's Develop It

Add layout positioning!

Absolutely position an image to the  
top right

Add a header, two columns and a  
footer through floating

# Questions?





Topom  
down here