



FEWD - Layout

James Gallichio

Developer, Plattar

Learning Objectives

- Use classes and IDs to apply styles to specific elements
- Apply header, footer, sidebar, and multi-column layouts to develop a web page
- Experiment and predict effects of floats and clearing CSS positioning

Agenda

- Review
- HTML5 Structural Elements
- Divs, Classes and IDs
- Floats
- Flexbox
- Lab Time

Review

What would you like to review?

Div

```
<div>...</div>
```

Divs are the regular, go-to, generic "Make this a block of content!" tag to use.

HTML5 Block Elements

`<div>` is very generic and non-descriptive.

We can also add structure to HTML elements that are related to content layout.

- `<header>`
- `<main>`
- `<aside>`
- `<footer>`

class & id

With classes and ids we can target specific elements on a page, so we can manipulate it uniquely.

class & id



class & id

IDs are unique

Classes are not unique

class & id

How to select classes in CSS

```
.className
```

```
#idName
```

class & id

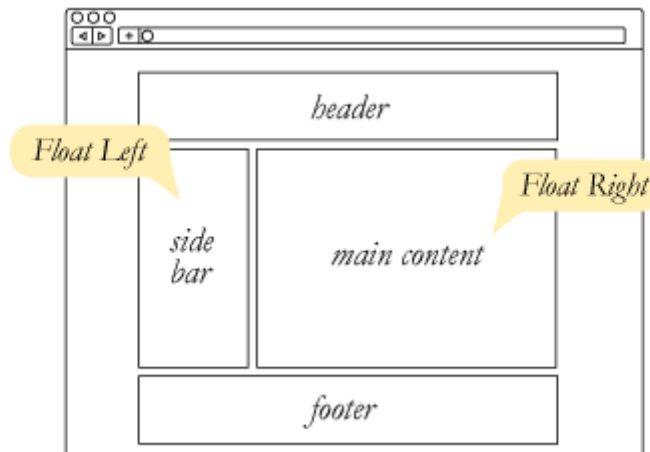
When should you use them?



class & id

Floats

Float is a CSS positioning property, originally intended to allow text and inline elements to flow around block content but often used for layouting.



Clearfix

Floated element heights don't get respected. But we can fix this by adding a new class to the parent of floated elements and some CSS:

CSS:

```
.clearfix:after {  
  content: " ";  
  display: block;  
  clear: both;  
}
```

HTML:

```
<div class="parentOfFloatedClass clearfix">  
  <p class="floatedElement"></p>  
</div>
```



Floating Layouts

Other solutions

- `display: inline-block;`
- Flexbox

Flexbox

```
.containerElement {  
display: flex;  
}
```

Flexbox

- Prescribe layout in the container element as much as possible

```
.containerElement {  
display: flex;  
justify-content: center;  
align-items: top;  
}  
  
.childElement1 {  
width: 30%;  
}
```

Flexbox

- But sometimes you want a child to behave differently

```
.containerElement {  
display: flex;  
justify-content: center;  
align-items: top;  
}  
  
.childElement1 {  
width: 30%;  
}  
  
.childElement3 {  
align-self: flex-end;  
  /* overrides parent align-items rule */  
}
```

Flexbox

For an indepth guide visit:

<https://css-tricks.com/snippets/css/a-guide-to-flexbox/>



Floating and Flexbox Layout



Layout Challenge

Further Reading

- CSS Flexbox
- Wireframing
- <http://caniuse.com>