CSS floats

- In CSS content “flows” in accordance with text direction
- In the case of inline elements, this can be imagined as words flowing on a page (link)

- Remember in addition to text. Images, video, audio, and inline-frames (iframe) are all inline elements

elements of unequal height

- When dealing with textual content – each element or character in a given block is usually of equal height
- However, when dealing with images or other (inline) elements – those elements may (and often should) be unequal in height to the surrounding content
- This may cause weird anomalies (link)

vertical align

- Sometimes, it is enough to realign the content – this can be done using the vertical-align CSS property

img {vertical-align: bottom;}

- Commonly used values are: top, bottom, and middle

- However, this is often not a viable solution: top, bottom, middle

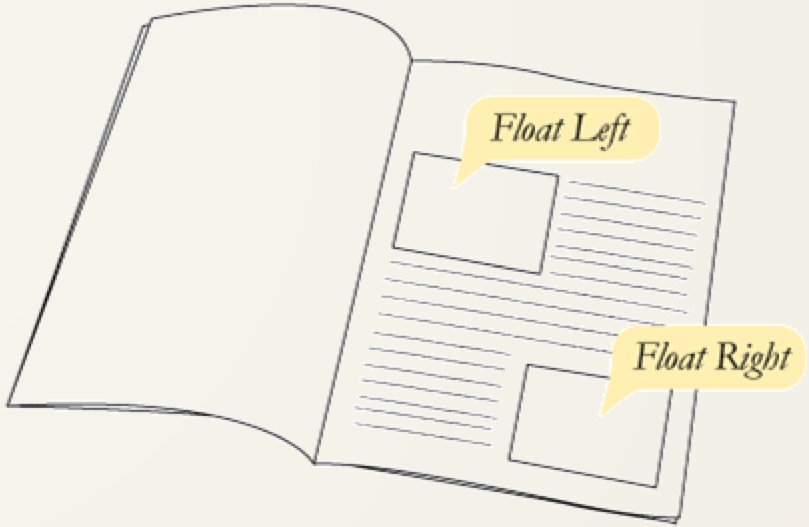
images outside of text blocks

- One solution is to make sure the offending element (usually an image) is placed between block elements in the flow
- This results in the outside elements flowing vertically (link)

floats

- In print – we can often find images that are wrapped by textual content
- In print this is called text wrap, and the image can be centered or justified

- In CSS we can use the float property to create a justified text wrap effect



- the float property is used
- It can have one of two values: left, and right

img {float:left;}

- Floated elements will be “pushed” in the float direction, as much as possible – until reaching the border of the parent element
- Floated elements will float past elements that are on the same line as the floated element, as well as any elements which are placed after it in the flow (link)

- Most inline elements have no margins, this causes text that wraps around floating elements “cling” to the floated element, often impacting legibility

- Manually adding margins can resolve the issue (link)
- To keep layout consistency – asymmetrical margins can be used (link)

multiple floating elements

- When multiple elements are floated, they act as if they are part of a separate flow (link)

- There are two float flows, left and right
- You can imagine that the float value is the "from" text direction as in "from **left**-to-right" or "from **right**-to-left"

floating blocks

- When floating a block element, there is usually no visible effect – because by default the element will occupy the entirety of it's horizontal line
- If you set a width to a block, it will float as if it were an inline element: link

- Note that the block will act similarly to an inline element, but it's content will not float

clearing floats

- Imagine that floating content is part of a separate (but not isolated) text flow
- Like with inline elements, floating elements will keep floating one after another, until they reach the edges of the parent element (link)

- The clear property may be used to “begin an new line” of floating elements
- Clear accepts values of left, right, or both

```
.clear { clear:both;}
```

- Use an element with a class to clear floats (link)

```
<span class=“clear“></span>
```