COMP2155

BUILDING WEB APPLICATIONS

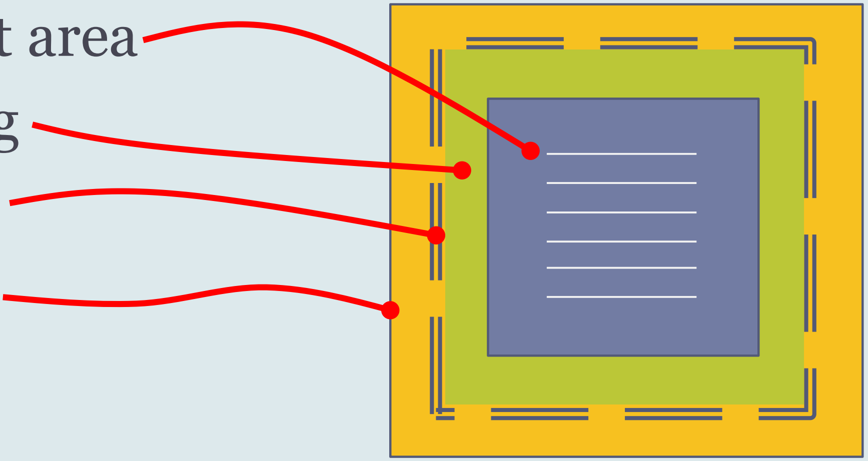
DR. CURTIS GITTENS

LECTURE 9

The CSS Box Model

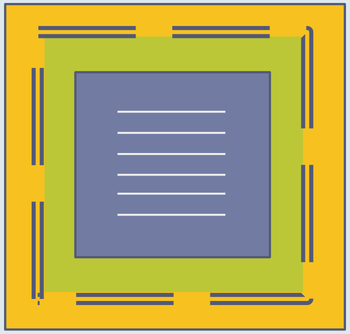
The Box Model

- CSS sees the world as a box
- Every element in an HTML document has a box made up of four parts:
  - ○ Content area
  - ○ Padding
  - ○ Border
  - ○ Margin



The Box Model

- Being the box
  - Content: holds the content (text or an image, for instance)
  - Padding: optional and transparent
  - Border: optional and can be placed around the padding using various styles
  - Margin: optional and transparent. The margin surrounds everything
  - All elements are treated as boxes: paragraphs, headings, block quotes, lists, list items, etc.
  - Inline elements like <em> and links also



The Box Model

- Using the box
  - ○ Given the rule below, describe what the box would look like?

```
.theboxstyle{
  font-family: Verdana, “Times New Roman”, serif;
  color: #444444;
  border-color: black;
  border-width: 1px;
  border-style: solid;
  background-color: #a7cece;
  padding: 25px;
  padding-left: 80px;
  margin: 30px;
  margin-right: 250px;
}
```

Using CSS Shorthand

- Padding Shorthand
  - Padding has four properties: padding-top, padding-right, padding-bottom and padding-left
  - The short form for specifying padding is:

```
padding: 0px 20px 20px 40px;  /* order is specific */
padding: 35px;
padding: 0px 20px
```

- Margin Shorthand
  - Margin has the same: margin-top, margin-right, margin-bottom, margin-left

```
margin: 0px 20px 20px 40px;  /* order is specific */
margin: 40px;
margin: 0px 40px
```

Using CSS Shorthand

- Border Shorthand
  - Border shorthand is similar: border-width, border-style, border-color

```
border-width: thin;
border-style: solid;
border-color: #007e7e;
border: solid #007e7e  thin  /* order independent */
```

- Background Shorthand
  - Same type of order independence with different properties

```
border-color: white;
border-image: url(images/myportrait.jpg);
border-repeat: repeat-x;
background: repeat-x white url(images/myportrait.jpg);
```

...and now for something COMPLETELY DIFFERENT

Block-level Boxes

- Boxes that participate in a **block formatting context**
- Block-level elements generate a *principal block-level box*
  - Contains descendant boxes and generated content
  - The box involved in any positioning scheme.
  - Additional boxes besides the principal box can also be generated by some block-level elements
    - e.g. 'list-item' elements.
    - These additional boxes are placed with respect to the principal box.

Block Formatting Context

- Boxes laid out vertically one after the other
- Begins at the top of a containing block
- The vertical distance between two sibling boxes is determined by the 'margin' properties.
- Vertical margins between adjacent block-level boxes in a block formatting context **collapse**
- Each box's left outer edge touches the left edge of the containing block
  - Even when floats are present, but line boxes (text) may shrink
    - A **new formatting context** would stop shrinkage, but box itself may become narrower

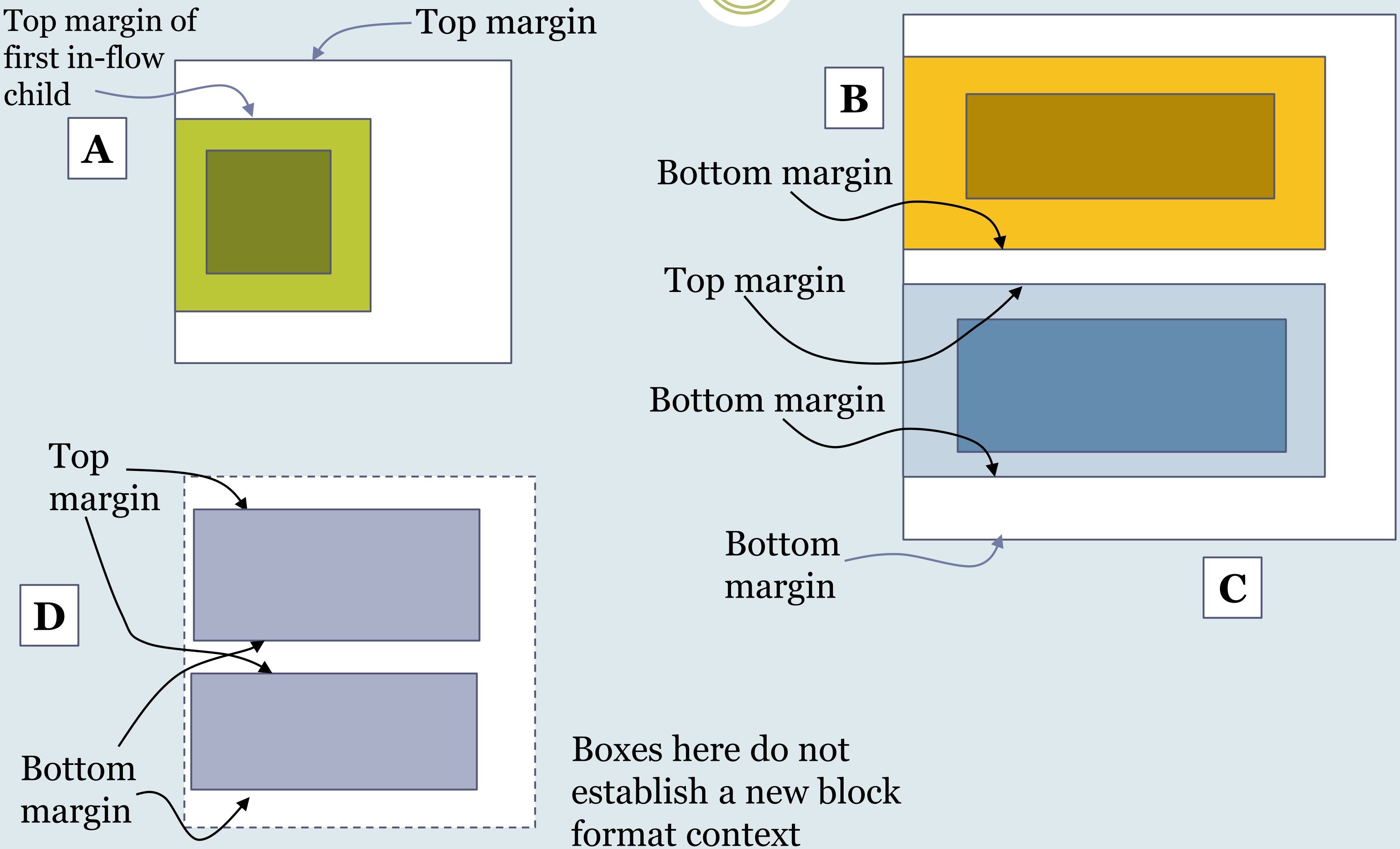
Block Formatting Context

- The following document objects create new formatting contexts for their contents
  - Floats
  - Absolutely positioned elements
  - Block containers
    - Inline-blocks, table-cells, and table-captions that are not block boxes
  - Block boxes with 'overflow' other than 'visible'
    - Except when that value has been propagated to the viewport

Adjoining Margins

- Two margins are adjoining if and only if:
  - Both belong to in-flow **block-level boxes** that participate in the same **block formatting context**
  - No line boxes, no clearance, no padding and no border separate them
  - Both belong to vertically-adjacent box edges, i.e. form one of the following pairs:
    - **A** Top margin of a box and top margin of its first in-flow child
    - **B** Bottom margin of box and top margin of its next in-flow following sibling
    - **C** Bottom margin of a last in-flow child and bottom margin of its parent
      - The parent must have a computed height of ‘auto’
    - **D** Top and bottom margins of a box that does not establish a new block formatting context
      - Must have zero computed 'min-height', zero or 'auto' computed 'height', and no in-flow children

Vertically-adjacent Box Edges



Boxes here do not establish a new block format context

Collapsing Margins

- Adjoining margins of two or more boxes can combine to form a single margin.
  - These margins are said to collapse
- The resulting combined margin is called a *collapsed margin*
- Exceptions to adjoining vertical margins collapse:
  - Margins of the root element's box do not collapse.
  - If the top and bottom margins of an element with clearance are adjoining, its margins collapse with the adjoining margins of following siblings but that resulting margin does not collapse with the bottom margin of the parent block.
- Horizontal margins never collapse.

Collapsing Margin Implications

- Margins between a floated box and any other box do not collapse
  - Not even between a float and its in-flow children
- Margins of elements that establish new block formatting contexts do not collapse with their in-flow children.
  - e.g. floats and elements with 'overflow' other than 'visible'
- Margins of absolutely positioned boxes do not collapse (not even with their in-flow children).
- Margins of inline-block boxes do not collapse (not even with their in-flow children).

Collapsing Margins Implications

- The bottom margin of an in-flow block-level element always collapses with the top margin of its next in-flow block-level sibling
  - The sibling cannot have clearance, i.e. clear property not ‘none’
- The top margin of an in-flow block element collapses with its first in-flow block-level child's top margin if:
  - The element has no top border, no top padding, and the child has no clearance.

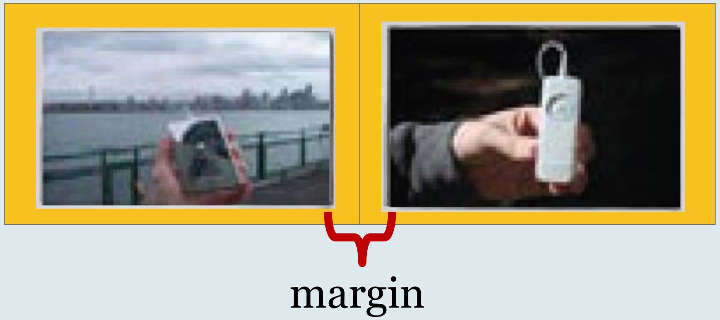
Collapsing Margins Implications

- The bottom margin of an in-flow block box collapses with its last in-flow block-level child's bottom margin if:
  - Block box has 'height' of 'auto' and a 'min-height' of zero, no bottom padding and no bottom border
  - The child's bottom margin does not collapse with a top margin that has clearance.
- A box's own margins collapse if:
  - the 'min-height' property is zero,
  - it has neither top or bottom borders nor top or bottom padding,
  - has a 'height' of either 0 or 'auto', and it does not contain a line box, and all of its in-flow children's margins (if any) collapse.

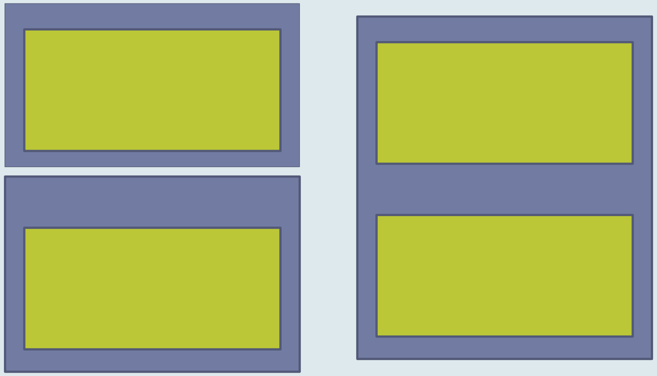
Collapsing Margins Summary

- Inline elements next to each other
  - Margins of the two inline elements are **added together** and the browser provides the space
- Block elements on top of each other
  - Margins are collapsed with the resulting height being the height of the largest margin

Inline elements



Block-level elements

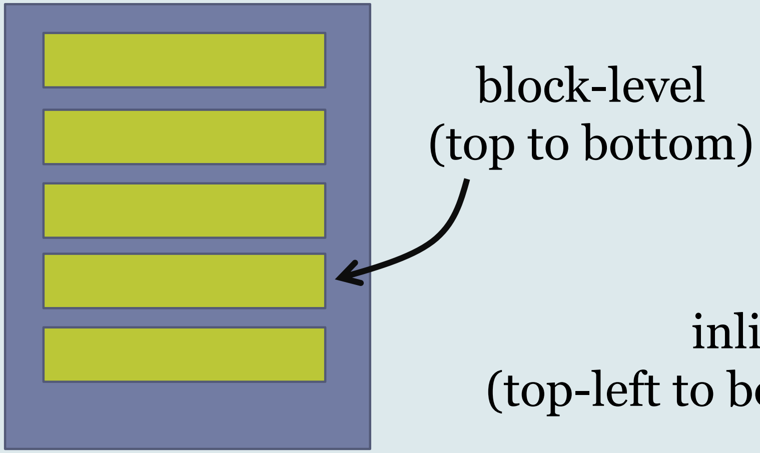


…and now, back to our show

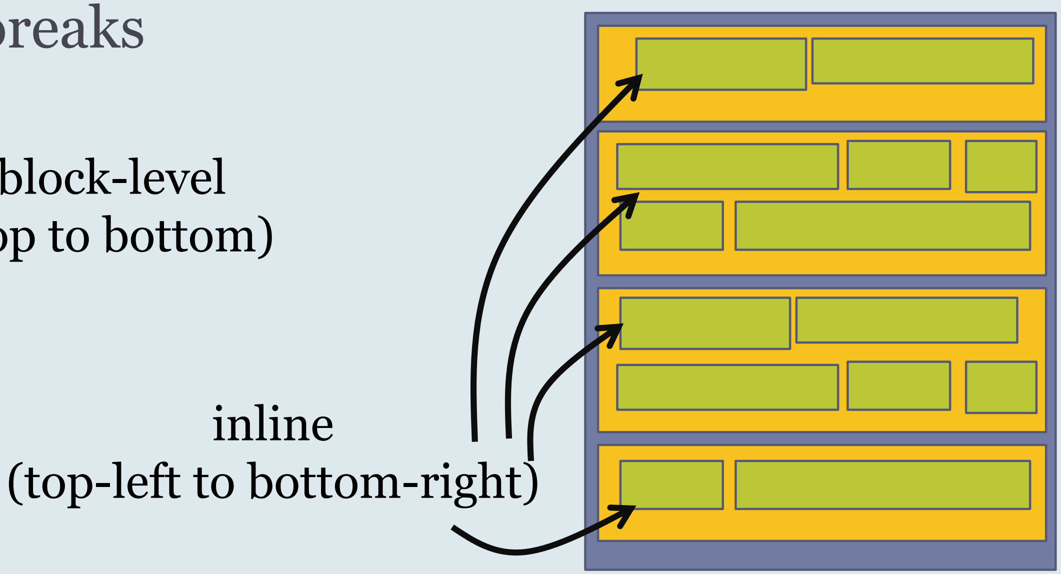
Page Flow

Content Flows Down Page

- CSS lays out (flows) the content on the page vertically – from top to bottom
  - With block-level elements, a line-break is inserted after each one
  - With inline elements, content flows from top left to bottom right without line breaks



block-level
(top to bottom)



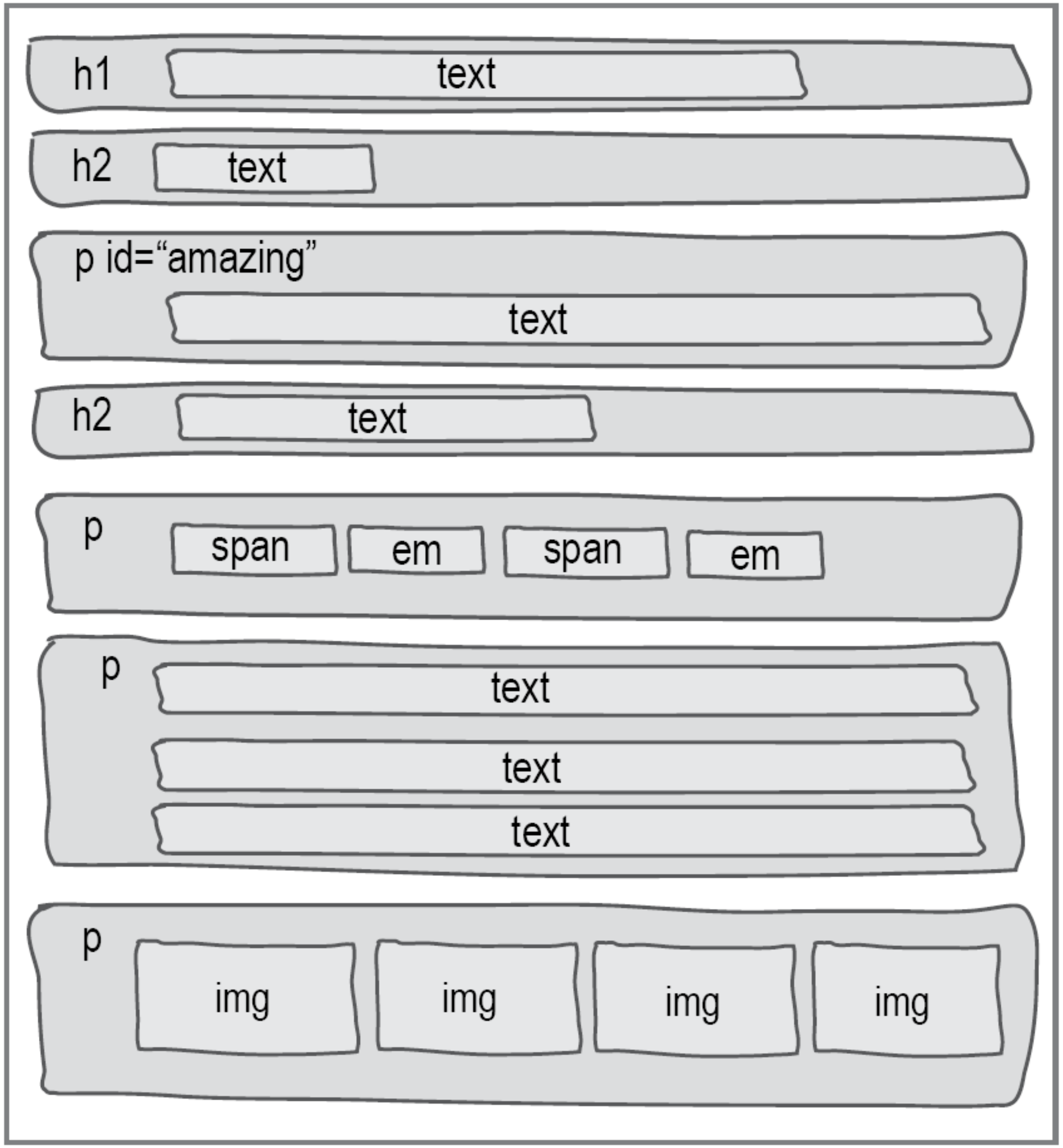
inline
(top-left to bottom-right)

Content Flows Down Page

The element to be floated is assigned an id

It is resized (optional)

Then the float property is assigned to the style

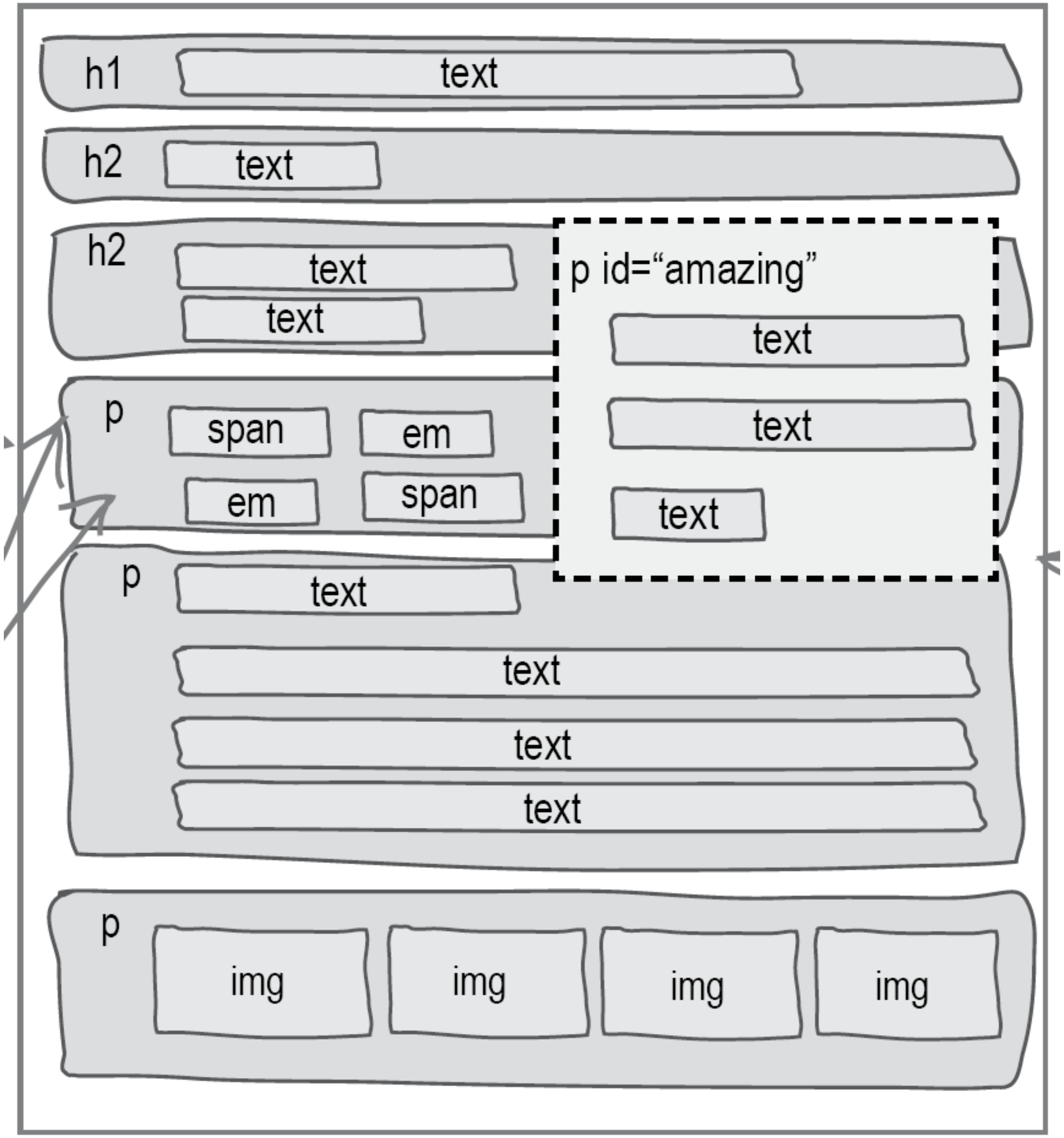


Content Flows Down Page

The float property causes the browser to float all the elements, minus the floated element, as usual. Then it places the floated element beneath the block element where it was located.

Block level elements remain in the same order – ignoring the floated element

Inline elements recognize the floated element



Content Flows Down Page

- Absolutely Positioned elements are…
  - Removed completely from the flow of the page
  - Placed in the position indicated by the top and right properties (or bottom and left as well)
  - Unknown to the other elements in the flow of the page
    - Inline elements do not flow around absolutely positioned elements
  - Can be layered on top of each other
    - The z-index property is used to determine which element is on top
  - Ignored by the clear property that ensures no elements appear to the left or right of an element
  - Specified by the position relative to the closest ancestor element that is also positioned

Content Flows Down Page

- Fixed Position elements:
  - Are positioned by the offset from the edge of the browser’s window rather than the page
  - Stays where placed even if the page is scrolled
- Relative Position elements
  - Not used frequently
  - Still part of the flow of the page, but at the last moment, just before the element is displayed, the browser offsets its position
    - This allows space around the element to be maintained since inline elements respects its margins
    - The element can be shifted within its allotted space

Content Flows Down Page

- In Summary
  - Block elements ignore both floated and relative positioned elements
  - Inline elements respect margins of both floated and relative positioned elements
  - All elements ignore both absolutely positioned and fixed position elements
  - Order of <div>s is important since some user agents display them in the order they appear

Collapsed Margins Examples

- http://reference.sitepoint.com/css/collapsingmargins
- http://www.howtocreate.co.uk/tutorials/css/margincollapsing