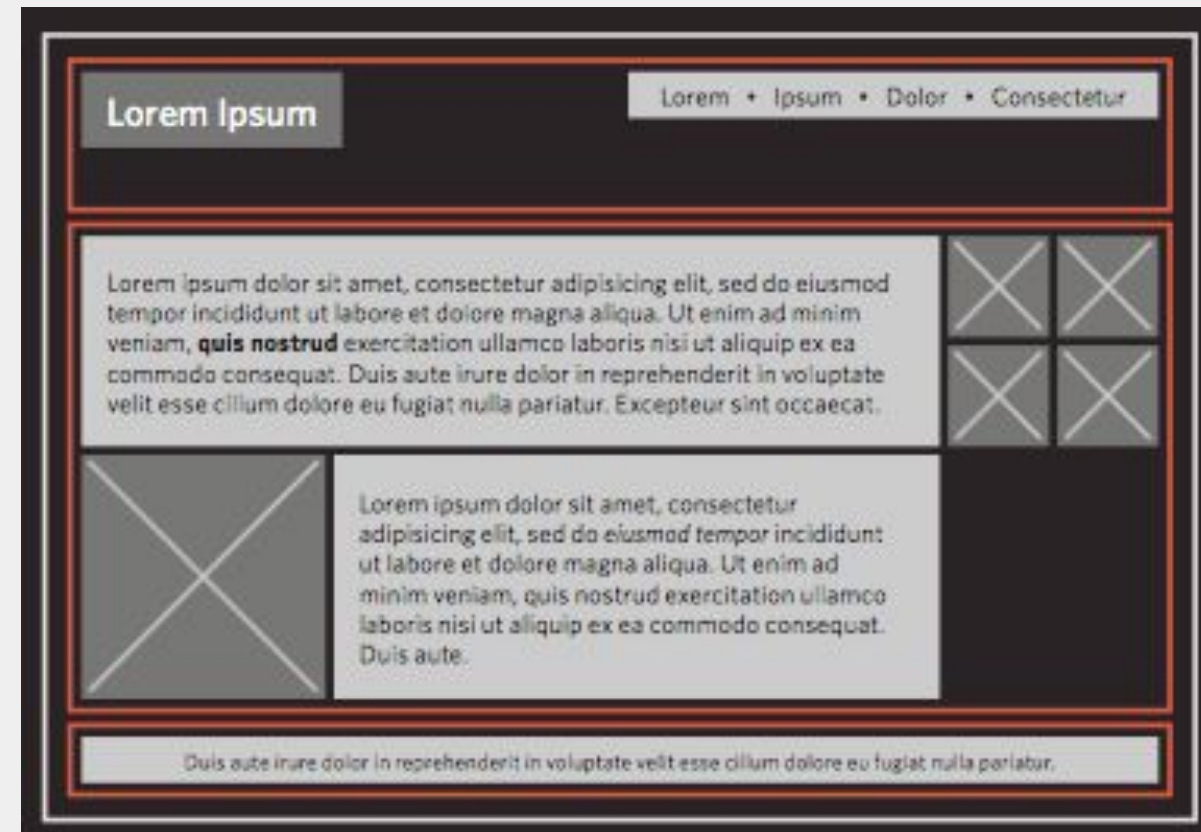# BIT

Institute of
Technology
Belgrade

BIT

Layout

Building Blocks

- CSS treats each HTML element as if was is in its own box. This box will either be a **block-level** box or an **inline** box.

- Block-level elements
  - Start on a new line
  - *<h1>, <p>, <ul>, *

- Inline elements
  - Flow in between surrounding text
  - <img>, <b>, <i>

Containing Elements

BIT

- If one block-level element sits inside another block-level element, the outer box is called a **containing** or **parent** element.

- It is common to group a number of elements together inside a **<div>**

- The orange lines in diagram represent **<div>** elements.

Controlling Position of Elements I

BIT

CSS has the following positioning schemes that allow you to control the layout of a page:

normal flow

Every block-level element appears in a new line, causing each item to appear lower down the page than the previous one.
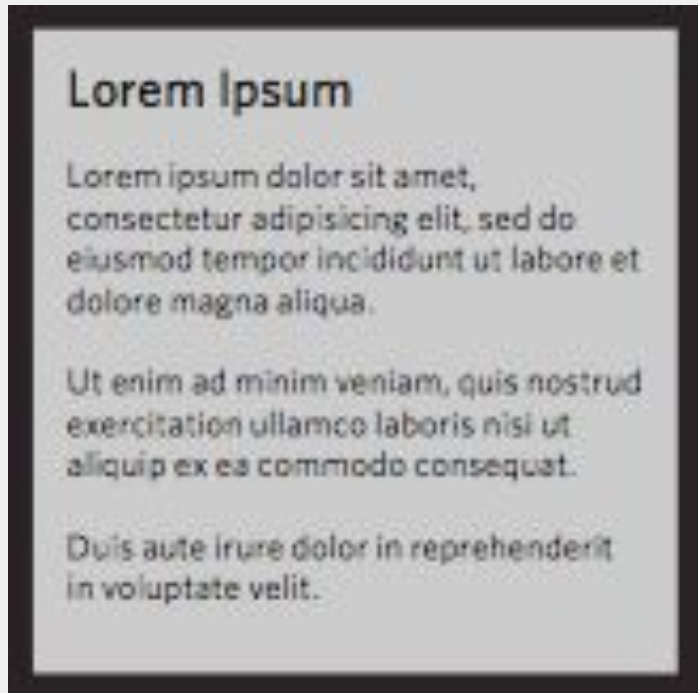
relative positioning

Moves an element from the position it would be in normal flow, shifting it to the top, right, bottom, or left of where it would have been placed.
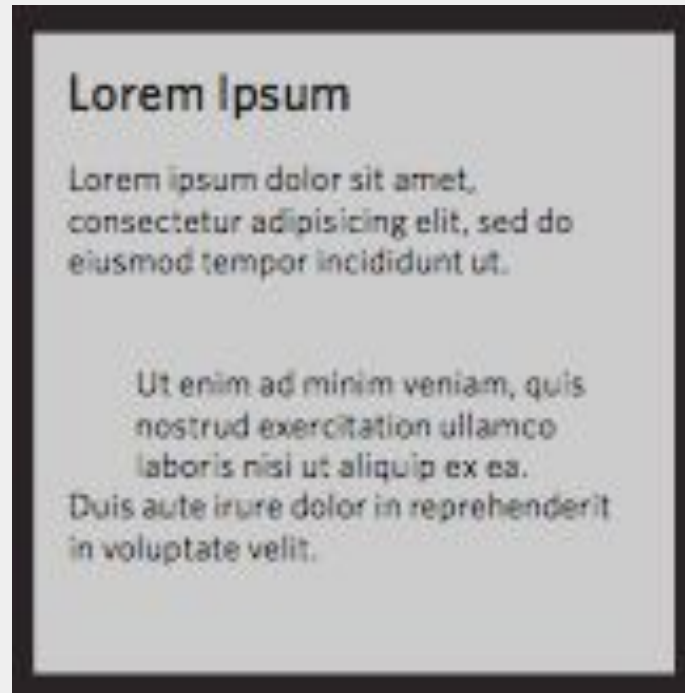
absolute positioning

Positions the element in relation to its containing element. It is taken out of normal flow, meaning that it does not affect the position of any surrounding elements.

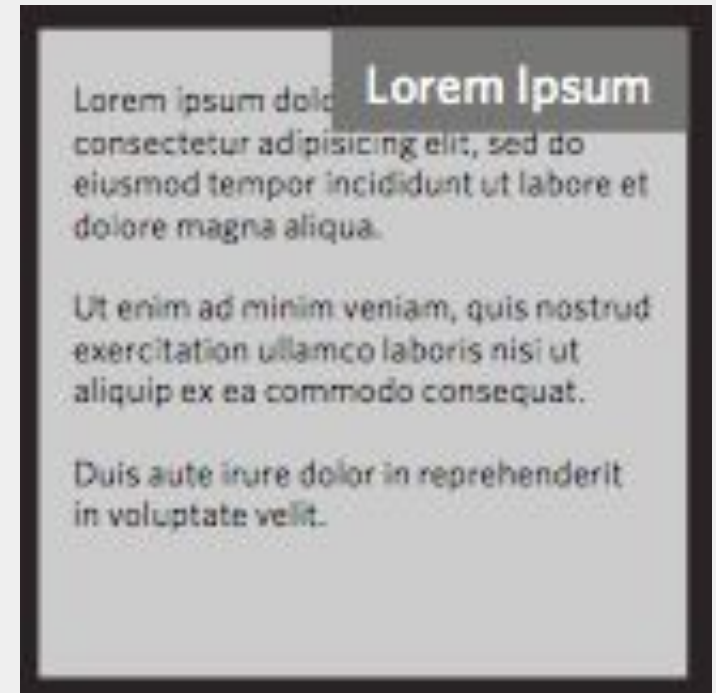# Controlling Position of Elements II

Normal flow

Relative position

Absolute position

BIT

Demo

Normal Flow

B I T

- To specify positioning (flow) of the elements on page, use the **position** property

- Default value of this property is **static** (normal flow)

- Each block-level element sits on top of the next one

- This is the default positioning for all elements

Relative Position

- Relative positioning moves an element in relation to where it would have been in normal flow

- To switch to this type of positioning, use **position** property with value **relative**

- Offset properties are used to indicate how far to move the element from where it would have been in normal flow
  - top
  - bottom
  - left
  - right

```
p.example {
    position: relative;
    top: 10px;
    left: 100px;
}
```

In 1817 Baron von Drais invented a walking machine that would help him get around the royal gardens faster: two same-size in-line wheels, the front one steerable, mounted in a frame upon which you straddled. The device was propelled by pushing your feet against the ground, thus rolling yourself and the device forward in a sort of gliding walk.

The machine became known as the Draisienne (or "hobby horse"). It was made entirely of wood. This enjoyed a short lived popularity as a fad, not being practical for transportation in any other place than a well maintained pathway such as in a park or garden.

The next appearance of a two-wheeled riding machine was in 1865, when pedals were applied directly to the front wheel. This machine was known as the velocipede (meaning "fast foot") as well as the "bone shaker," since it's wooden structure combined with the cobblestone roads of the day made for an extremely uncomfortable ride. They also became a fad and indoor riding academies, similar to roller rinks, could be found in large cities.

In 1817 Baron von Drais invented a walking machine that would help him get around the royal gardens faster: two same-size in-line wheels, the front one steerable, mounted in a frame upon which you straddled. The device was propelled by pushing your feet against the ground, thus rolling yourself and the device forward in a sort of gliding walk.

The machine became known as the Draisienne (or "hobby horse"). It was made entirely of wood. This enjoyed a short lived popularity as a fad, not being practical for transportation in any other place than a well maintained pathway such as in a park or garden.

The next appearance of a two-wheeled riding machine was in 1865, when pedals were applied directly to the front wheel. This machine was known as the velocipede (meaning "fast foot") as well as the "bone shaker," since it's wooden structure combined with the cobblestone roads of the day made for an extremely uncomfortable ride. They also became a fad and indoor riding academies, similar to roller rinks, could be found in large cities.

Absolute Position I

- Use the **position** property with value **absolute** to position element in this way

- The box is taken out of normal flow and no longer affects the position of other elements on the page

- Element is positioned relative to its first positioned (not static) ancestor element; if it has no such ancestor element it is positioned relative to viewport

- Use offset properties to specify exact position

Absolute Position II


```
h1 {
    width: 200px;
    background: ▉pink;
}
```

```
h1 {
    position: absolute;
    right: 10px;
    top: 100px;
    width: 200px;
    background: ▉pink;
}
```

Fixed Positioning I

- Fixed positioning is a type of absolute positioning

- Uses the **position** property with the **fixed** value

- Positions the element in relation to the browser window

- Element stays in the exact same place while user scrolls the page

- Box offset properties are used to define where should the box appear inside the browser window
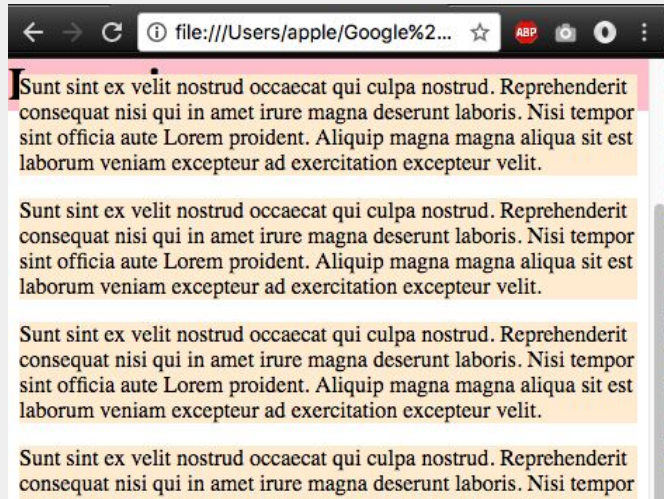
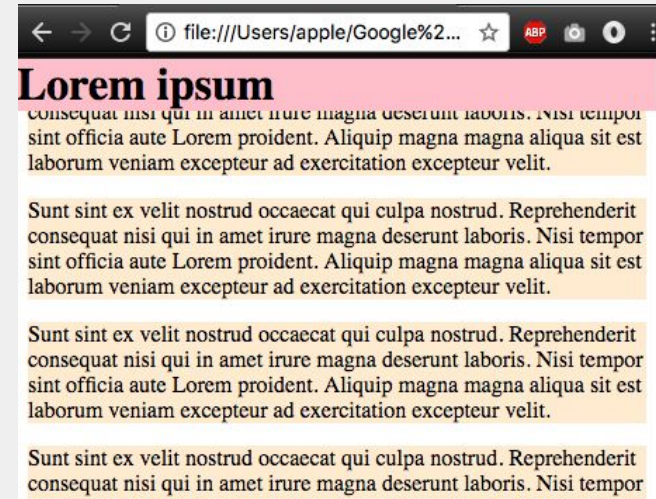# Fixed Positioning II

BIT

Exercise

Overlapping Elements I

- When using relative, fixed, or absolute positioning, boxes can overlap
- Elements that appear later in the HTML code sit on top of those that are earlier in the page
- Elements with non-static positioning will always appear on top of elements with default static positioning.
- To control which element sits on top, use the **z-index** property
- **z-index** only effects elements that have a position value other than static (the default).
- Value that can be assigned to this property is a number
- Element with a greater number will be in front of the one with the lower number
- Similar feature as "bring to front" in PowerPoint

Overlapping Elements II


```
h1 {
    position: fixed;
    top: 0;
    left: 0;
    width: 100%;
    background: ■pink;
    margin: 0;
}
p {

    position: relative;
    top: 40px;
    background: ■blanchedalmond;
}
```

```
h1 {
    z-index: 1;
    position: fixed;
    top: 0;
    left: 0;
    width: 100%;
    background: ■pink;
    margin: 0;
}
p {

    position: relative;
    top: 40px;
    background: ■blanchedalmond;
}
```

Exercise

Floating Elements I

- Places the element as far to the left or right of the containing element as possible

- Use **float** property with values **left** or **right**

- Everything inside the containing element will flow around the element that is floated

- If an empty element contains floated elements only, its height will be 0. This can be fixed by setting its `overflow` to `hidden`.

B I T

Lorem ipsum

Sunt sint ex velit nostrud occaecat qui culpa nostrud. Reprehenderit consequat nisi qui in amet irure magna deserunt laboris. Nisi tempor sint officia aute Lorem proident. Aliquip magna magna aliqua sit est laborum veniam excepteur ad exercitation excepteur velit.

Sunt sint ex velit nostrud occaecat qui culpa nostrud. Reprehenderit consequat nisi qui in amet irure magna deserunt laboris. Nisi tempor sint officia aute Lorem proident. Aliquip magna magna aliqua sit est laborum veniam excepteur ad exercitation excepteur velit.

Sunt sint ex velit nostrud occaecat qui culpa nostrud. Reprehenderit consequat nisi qui in amet irure magna deserunt laboris. Nisi tempor sint officia aute Lorem proident. Aliquip magna magna aliqua sit est laborum veniam excepteur ad exercitation excepteur velit.

Sunt sint ex velit nostrud occaecat qui culpa nostrud. Reprehenderit consequat nisi qui in amet irure magna deserunt laboris. Nisi tempor sint officia aute Lorem proident. Aliquip magna magna aliqua sit est laborum veniam excepteur ad exercitation excepteur velit.

```
span {
    border: 1px solid  #665544;
    background:  blanchedalmond;
}
```

Lorem ipsum

Sunt sint ex velit nostrud occaecat qui culpa nostrud. Reprehenderit consequat nisi qui in amet irure magna deserunt laboris. Nisi tempor sint officia aute Lorem proident. Aliquip magna magna aliqua sit est laborum veniam excepteur ad exercitation excepteur velit.

Sunt sint ex velit nostrud occaecat qui culpa nostrud. Reprehenderit consequat nisi qui in amet irure Nisi tempor sint officia aute Lorem proident. Aliquip magna magna aliqua sit est laborum veniam excepteur ad exercitation excepteur velit. Reprehenderit consequat nisi qui in amet irure magna deserunt laboris.

Sunt sint ex velit nostrud occaecat qui culpa nostrud. Reprehenderit consequat nisi qui in amet irure magna deserunt laboris. Nisi tempor sint officia aute Lorem proident. Aliquip magna magna aliqua sit est laborum veniam excepteur ad exercitation excepteur velit.
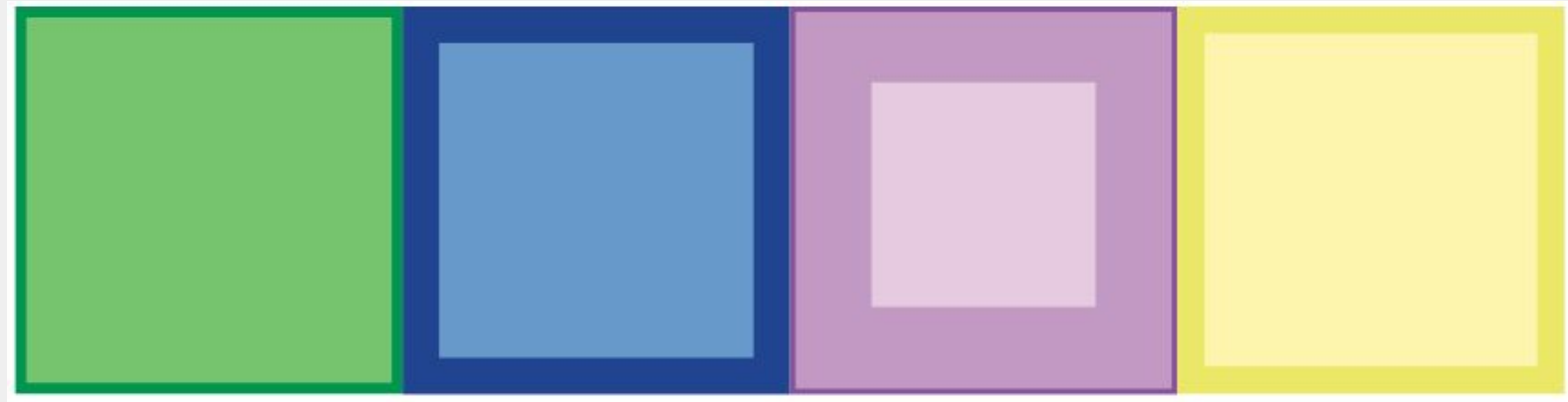
Sunt sint ex velit nostrud occaecat qui culpa nostrud. Reprehenderit consequat nisi qui in amet irure magna deserunt laboris. Nisi tempor sint officia aute Lorem proident. Aliquip magna magna aliqua sit est laborum veniam excepteur ad exercitation excepteur velit.

```
span {
    float: right;
    width: 300px;
    border: 1px solid  #665544;
    background:  blanchedalmond;
}
```

Floating Elements Problem

- When elements are floated, the height of the boxes can affect where the following elements sit

- Fourth paragraph has enough space to start under the third paragraph

Clearing Floats I

- The **clear** property allows you to say that no element should touch the left or right-hand sides of a box

- It can take the following values
  - **left**
  - **right**
  - **both**
  - **none**

Clearing Floats II


```
p {
    width: 300px;
    margin-right: 20px;
    float: left;
    background: ▩blanchedalmond;
}
```

```
p {
    width: 300px;
    margin-right: 20px;
    float: left;
    background: ▩blanchedalmond;
}
p.fourth {
    clear: left;
}
```

BIT

Exercise

References

- Floating: https://goo.gl/MgCfoP
- Float examples: https://goo.gl/MGVThb
- Box sizing: https://goo.gl/jFk6SY

BIT

Institute of
Technology
Belgrade