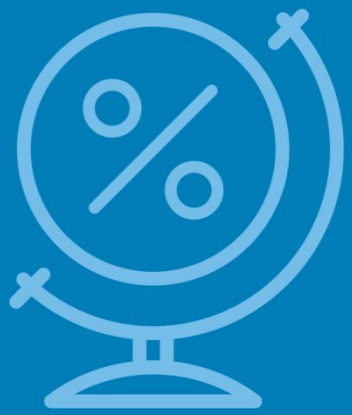
SoftServe | Empowering your Business through Software Development



Understanding CSS Essentials: Content Flow, Positioning, and Styling

Vyacheslav Koldovskyy
Last update: 12/01/2015

Agenda

- Presentation versus content
- CSS basics
  - The link between HTML and CSS
  - CSS selector and declaration
  - Fonts and font families
  - Web-safe fonts and @font-face rule
- Inline flow and block flow
- Float and absolute positioning
- Overflow

Presentation vs. Content

- Content is the words and images in an HTML document.
- Presentation is related to styles and how words and images "look" in an HTML document.
- Content is managed as HTML and style as CSS.
- The separation of HTML and CSS generally means keeping CSS styles in a file separate from the HTML file.

CSS

- CSS = Cascading Style Sheets
- CSS is a sequence of rules.
- CSS3 is the latest version, corresponds to HTML5
- CSS3 is that it's backward compatible with previous versions of CSS

How to add CSS to HTML?

1. Inlin

```
<h1                    argin-left:40px;">Some header</h1>
```

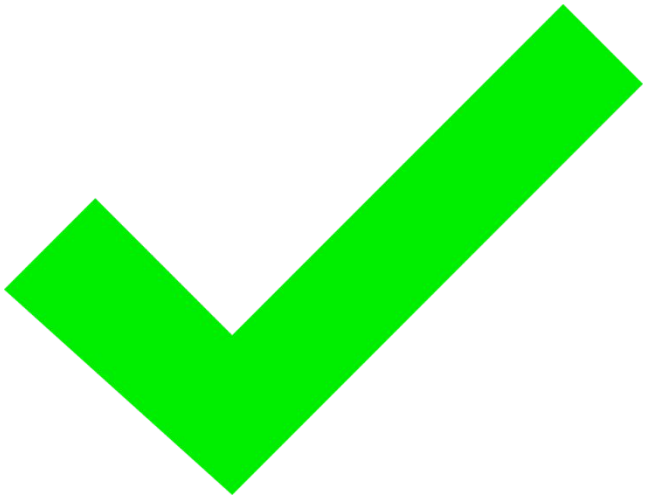
2. Style Shee

```
<
    yle>
    1 {
        color:
        margin-l                ;

 </st
</head>
```

3. External file:

```
<head>
  <link rel="stylesheet" href="default.css">
</head>
```



The Link Between HTML and CSS

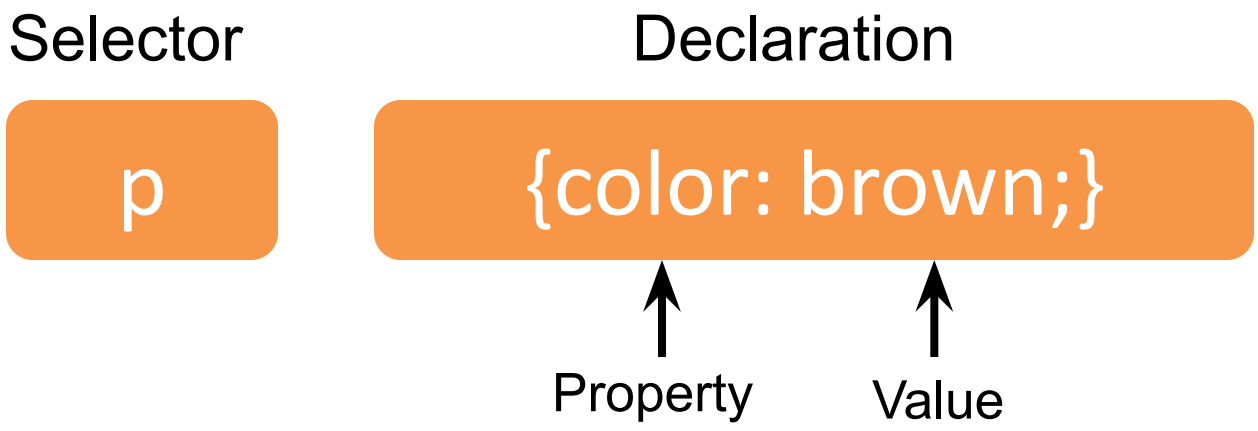
- The `<link>` element in an HTML file links the HTML file to a CSS file.
- You can reference more than one CSS file in an HTML page.
- Markup example:

```
<link href = "filename.css" rel =
"stylesheet" type = "text/css">
```

- For simple projects, can use the `<style>` tag to include styles within an HTML document

CSS Selector and Declaration

- The ***selector*** is usually the HTML element you want to style. The ***declaration*** is the style for a specific selector. A declaration has a property, which is a style attribute, and a value.



CSS selectors

- Basic Selectors
  - Type selectors `elementname`
  - Class selectors `.classname`
  - ID selectors `#idname`
  - Universal selectors `* ns|* *|*`
  - Attribute selectors `[attr=value]`
- Combinators
  - Adjacent sibling selectors `A + B`
  - General sibling selectors `A ~ B`
  - Child selectors `A > B`
  - Descendant selectors `A B`
- Pseudo-elements
- Pseudo-classes

https://developer.mozilla.org/en-US/docs/Web/CSS/Reference#Selectors

CSS Selectors – General

Selector	Example	Example description
.class	.intro	Selects all elements with class="intro"
#id	#firstname	Selects the element with id="firstname"
*	*	Selects all elements
element	p	Selects all <p> elements
element,element	div, p	Selects all elements and all <p> elements
element element	div p	Selects all <p> elements inside elements
element>element	div > p	Selects all <p> elements where the parent is a element
element+element	div + p	Selects all <p> elements that are placed immediately after elements
element1~element2	p ~ ul	Selects every element that are preceded by a <p> element

CSS Selectors – Attributes

Selector	Example	Example description
[attribute]	[target]	Selects all elements with a target attribute
[attribute=value]	[target=_blank]	Selects all elements with target="_blank"
[attribute~=value]	[title~=flower]	Selects all elements with a title attribute containing the word "flower"
[attribute|=value]	[lang|=en]	Selects all elements with a lang attribute value starting with "en"
[attribute^=value]	a[href^="https"]	Selects every <a> element whose href attribute value begins with "https"
[attribute$=value]	a[href$=".pdf"]	Selects every <a> element whose href attribute value ends with ".pdf"
[attribute*=value]	a[href*="w3schools"]	Selects every <a> element whose href attribute value contains the substring "w3schools"

CSS Pseudo-classes

- A CSS **pseudo-class** is a keyword added to selectors that specifies a special state of the element to be selected. For example :hover will apply a style when the user hovers over the element specified by the selector.
- Pseudo-classes, together with pseudo-elements, let you apply a style to an element not only in relation to the content of the document tree, but also in relation to external factors like the history of the navigator (:visited, for example), the status of its content (like :checked on some form elements), or the position of the mouse (like :hover which lets you know if the mouse is over an element or not). To see a complete list of selectors, visit CSS3 Selectors working spec.

Syntax

```
selector:pseudo-class {
  property: value;
}
```

CSS pseudo-elements

- Just like pseudo-classes, pseudo-elements are added to selectors but instead of describing a special state, they allow you to style certain parts of a document.
- For example, the ::first-line pseudo-element targets only the first line of an element specified by the selector.

```
selector::pseudo-element {
  property: value;
}
```

- All pseudo-elements

```
::after
::before
::first-letter
::first-line
::selection
::backdrop
```

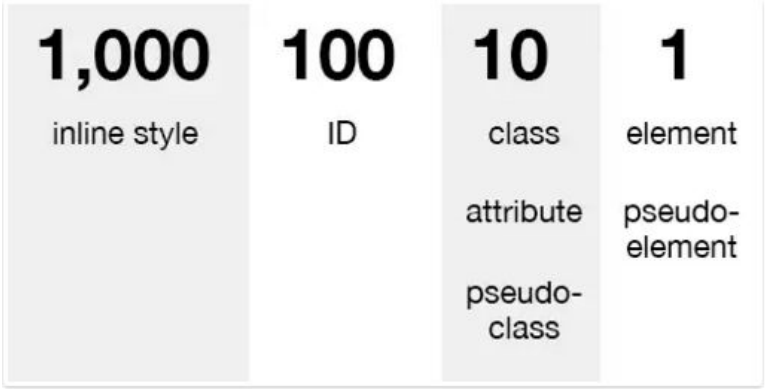
CSS Selectors – pseudo-classes and pseudo-elements

Selector	Example	Example description
:active	a:active	Selects the active link
::after	p::after	Insert something after the content of each <p> element
::before	p::before	Insert something before the content of each <p> element
:checked	input:checked	Selects every checked <input> element
:disabled	input:disabled	Selects every disabled <input> element
:empty	p:empty	Selects every <p> element that has no children (including text nodes)
:enabled	input:enabled	Selects every enabled <input> element
:first-child	p:first-child	Selects every <p> element that is the first child of its parent
::first-letter	p::first-letter	Selects the first letter of every <p> element
::first-line	p::first-line	Selects the first line of every <p> element
:first-of-type	p:first-of-type	Selects every <p> element that is the first <p> element of its parent
:focus	input:focus	Selects the input element which has focus
:hover	a:hover	Selects links on mouse over

Details: http://www.w3.org/TR/css3-selectors/

CSS Specificity

- Specificity is the means by which a browser decides which CSS property values are the most relevant to an element and therefore will be applied. Specificity is only based on the matching rules which are composed of css selectors of different sorts.
- The specificity is a weight that is applied to a given CSS declaration based on the count of each selector type. In the case of specificity equality, the latest declaration found in the CSS is applied to the element.
- Details: https://css-tricks.com/specifics-on-css-specificity/

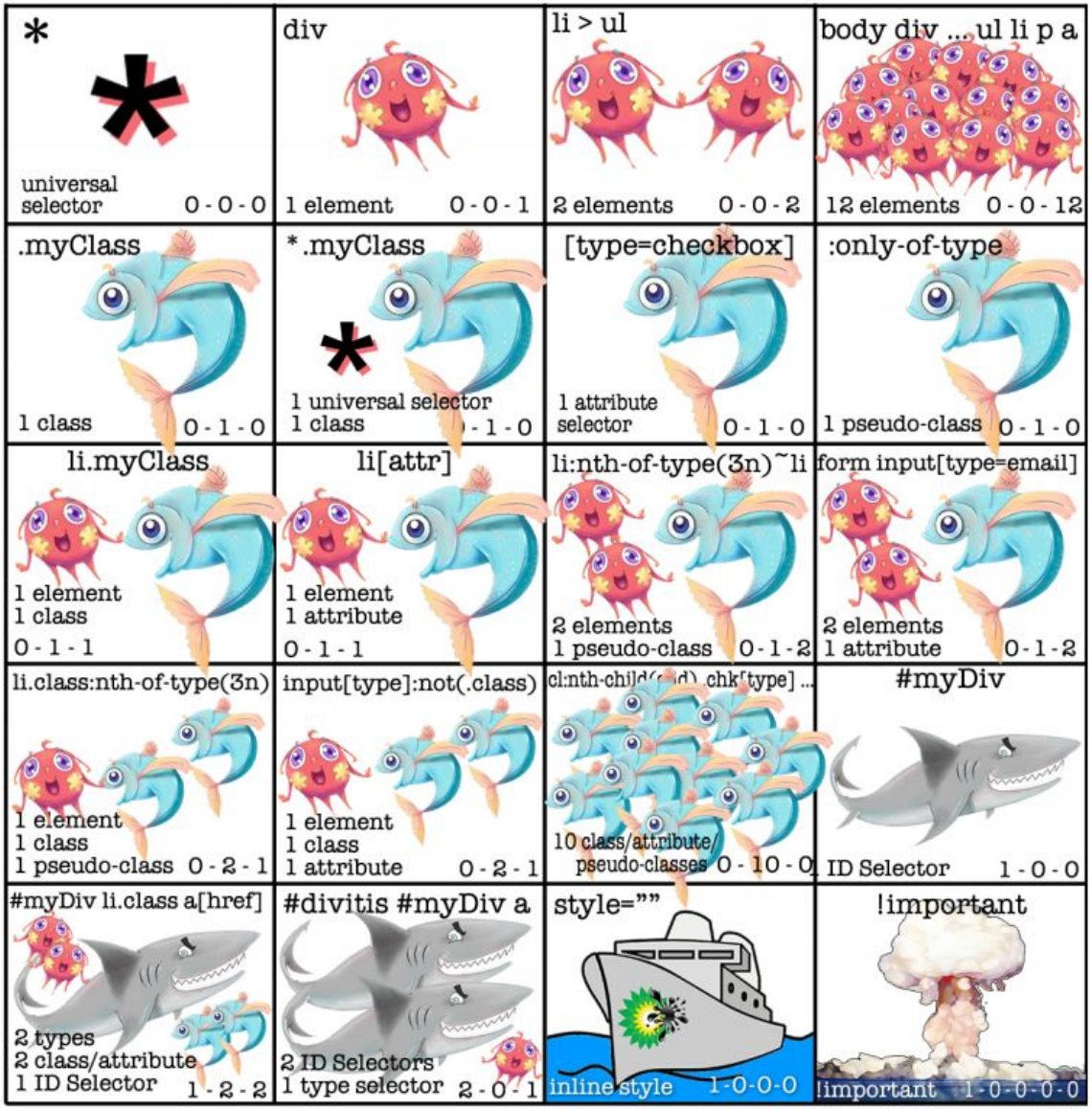


The !important exception

- When an **!important** rule is used on a style declaration, this declaration overrides any other declaration made in the CSS, wherever it is in the declaration list.
- Although, **!important** has nothing to do with specificity.
- Using **!important** is *bad practice* and should be avoided because it makes debugging more difficult by breaking the natural cascading in your stylesheets.

CSS SpeciFISHity

WITH PLANKTON, FISH AND SHARKS



ESTELLE WEYL * @ESTELLEVW * WWW.STANDARDISTA.COM * 2104

X-0-0: The number of ID selectors

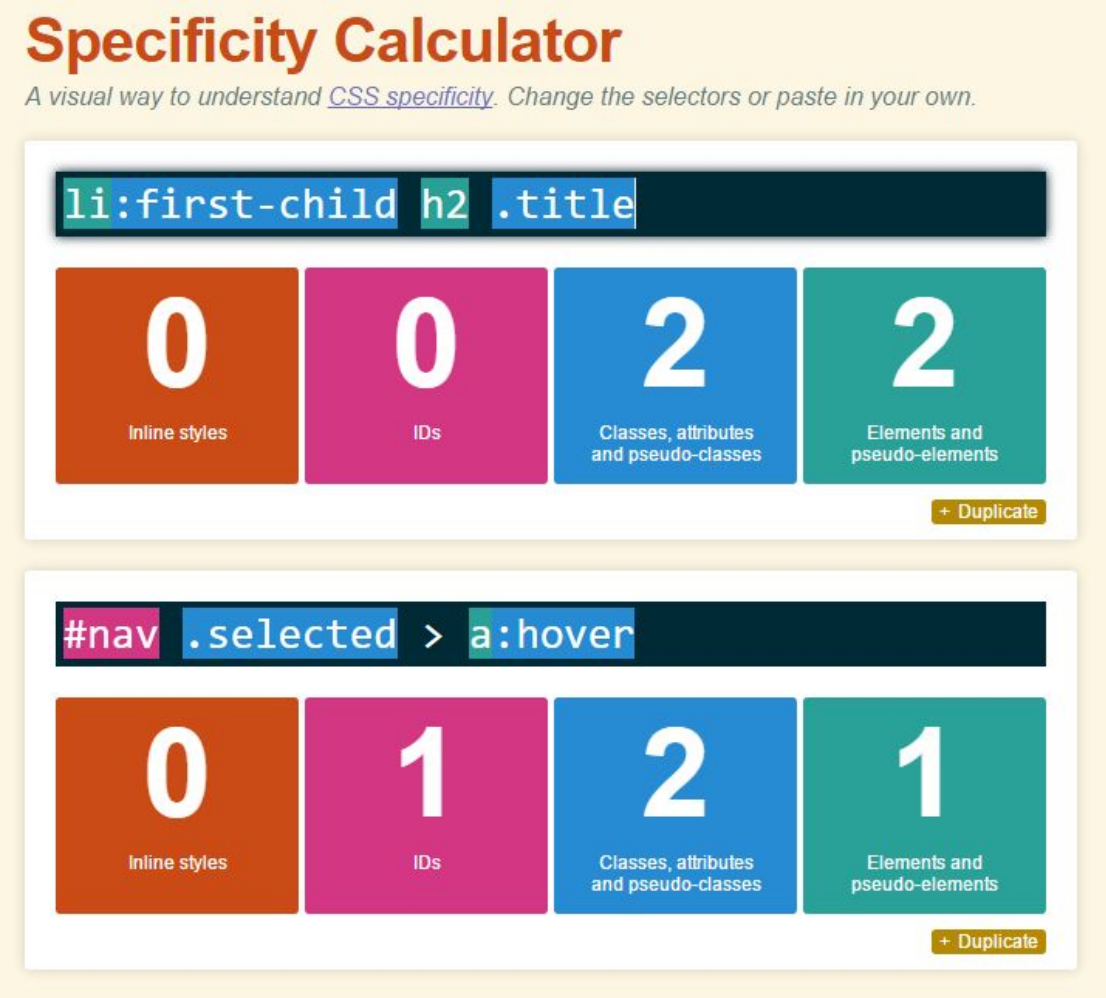
0-Y-0: The number of class selectors, attributes selectors, and pseudo-classes

0-0-Z: The number of type selectors and pseudo-elements

*, +, >, ~ : Universal selector and combinators do not increase specificity

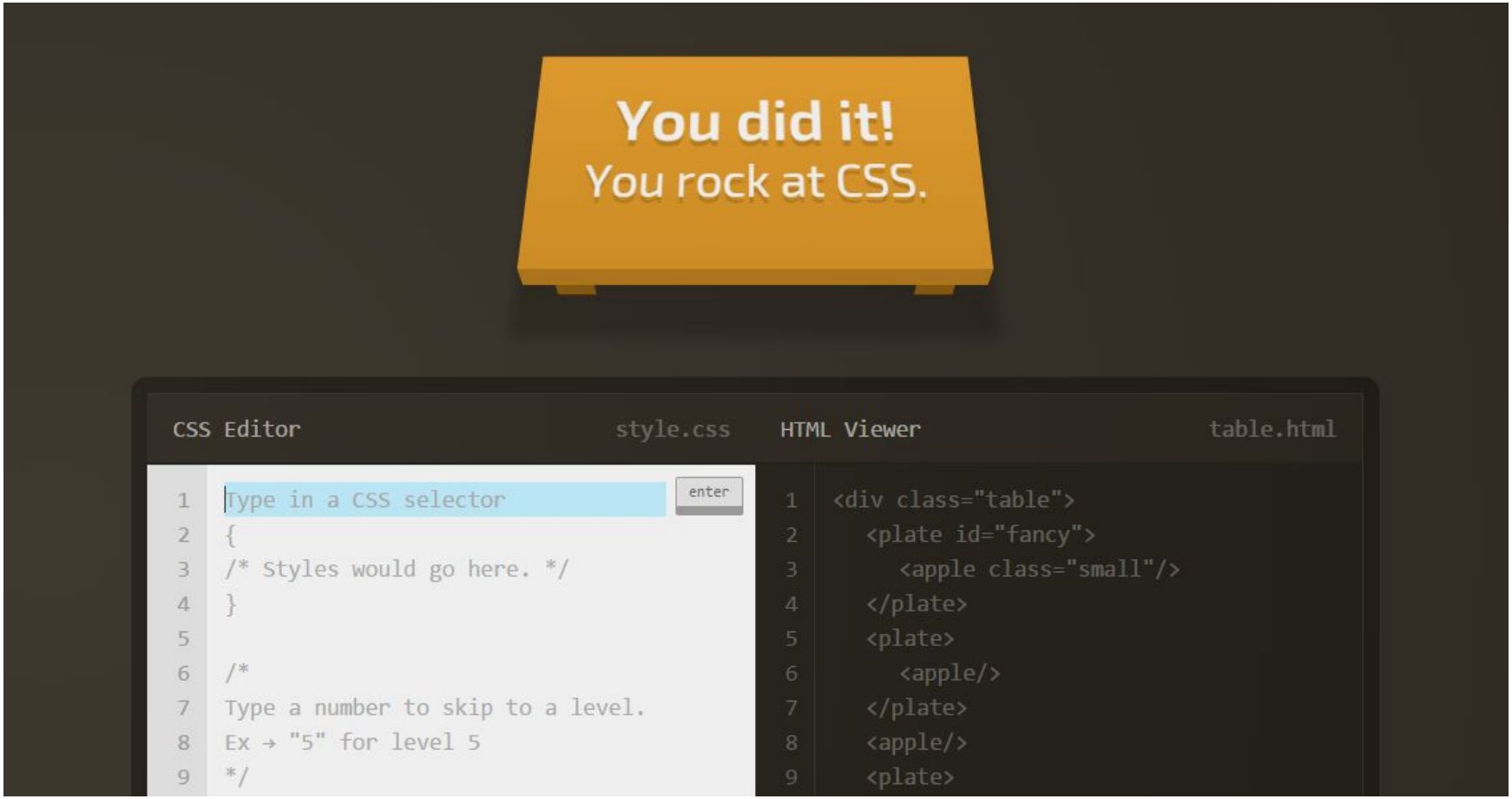
:not(x): Negation selector has no value. Argument increases specificity

CSS Specificity Calculator



http://specificity.keegan.st/

Practice Task: Pass CSS Game



http://flukeout.github.io/

Font Basics

- A ***font*** is a set of characters of a particular size and style.
- Examples:
  - Times New Roman
  - **Eras Bold ITC**
  - Myriad Web Pro
- `Monospace` is often used for technical material such as formulas, numbers, codes, and so on.

Serif and Sans Serif Fonts

Serif

Sans serif

Font Families

- The primary way to specify fonts in a CSS file is to use the `font-family` property.
- The property can declare a specific font, like Garamond or Arial, or a family that includes many different fonts, such as “serif.”
- Examples:
  - `font-family: Arial`
  - `font-family: serif`

Web-safe Fonts

- Fonts most likely installed on a Web page visitor's system
- List of Web-safe fonts is relatively short and doesn't offer much variety

`@font-face` Rule

- CSS3 rule that enables developers to use any font they choose
- Create a `font-face` rule by assigning a name to the font
- Font must be located on your Web server, or include a URL to font location
- Example:

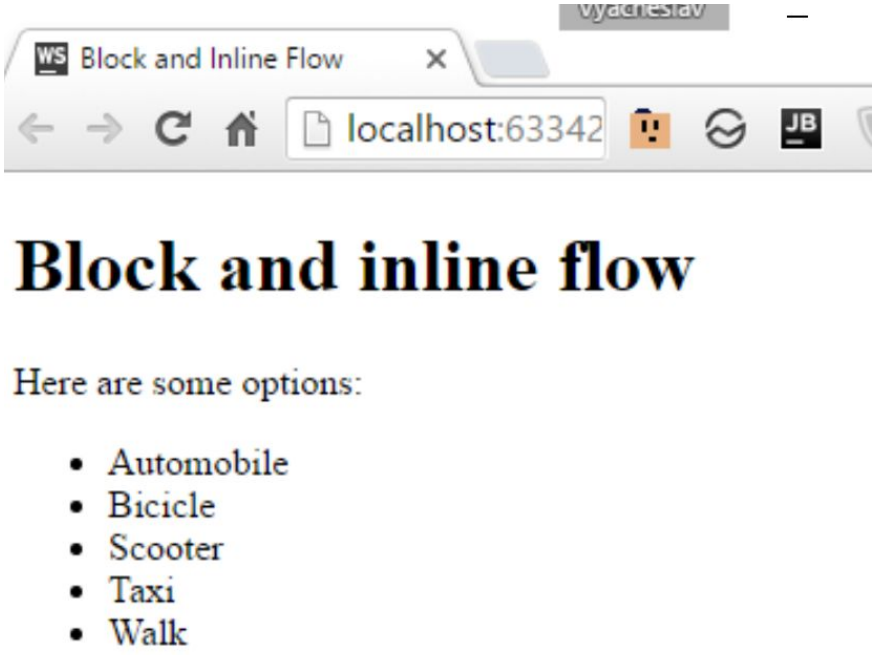
```
@font-face
{
font-family: TrustyHomePage;
src: url('Euphemia.ttf'),
}
```

Inline Flow and Block Flow

- Inline flow fills only as much width as required
- Block flow fills as much width as is available

Block Flow Example

```
<!doctype html>
<html>
    <head>
        <title>Block and Inline Flow</title>
    </head>
    <body>
        <h1>Block and inline flow</h1>
        <p>Here are som options:</p>
        <ul class="toolbar">
            <li>Automobile</li>
            <li>Bicicle</li>
            <li>Scooter</li>
            <li>Taxi</li>
            <li>Walk</li>
        </ul>
    </body>
</html>
```



Inline Flow Example

```
<!doctype html>
<html>
    <head>
        <title>Block and Inline Flow</title>
        <style>
            .toolbar li {
                display: inline;
                background-color: #EEE;
                border: 1px solid;
                border-color: #F3F3F3 #BBB #BBB #F3F3F3;
                margin: 2px;
                padding: .5em;
            }
        </style>
    </head>
    <body>
        <h1>Block and inline flow</h1>
        <p>Here are some options:</p>
        <ul class="toolbar">
            <li>Automobile</li>
            <li>Bicicle</li>
            <li>Scooter</li>
            <li>Taxi</li>
            <li>Walk</li>
        </ul>
    </body>
</html>
```

http://plnkr.co/edit/2ZQXJkbNMiqeV18n6pSV?p=preview

CSS Positioning

The **position** Property

The position property specifies the type of positioning method used for an element.

There are four different position values:

- static
- relative
- fixed
- absolute

Elements are then positioned using the top, bottom, left, and right properties. However, these properties will not work unless the position property is set first. They also work differently depending on the position value.

Z-Index: allows to place one element above another.

Details and samples:
http://www.w3schools.com/css/css_positioning.asp

Float Positioning

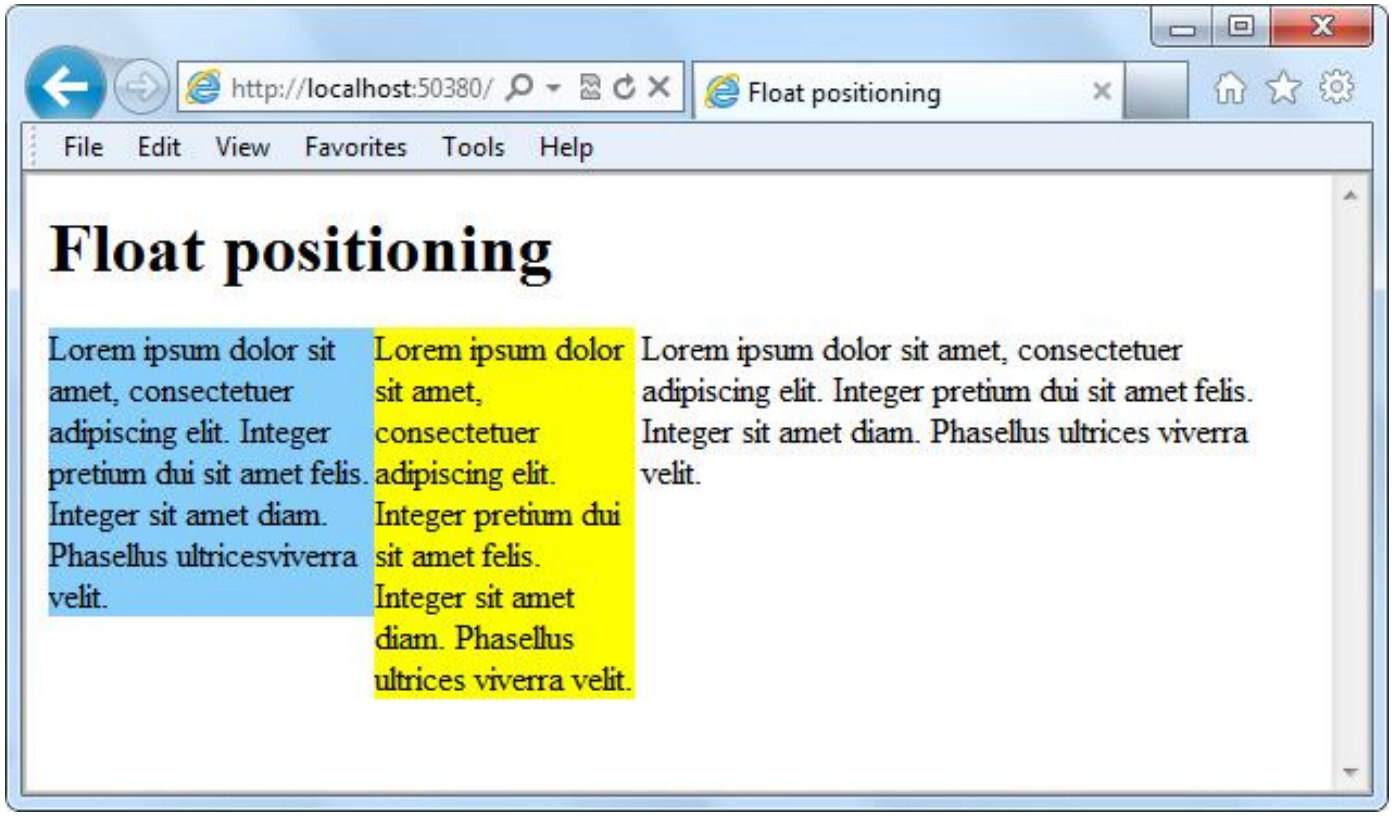
- Float positioning
  - Is useful when a layout is in columns, at least in part
  - To float an element is to have it move as far as possible either to the right or left
  - Text “wraps” around the element
- The **float** property specifies whether or not an element should float.
- The **clear** property is used to control the behavior of floating elements.

Float Positing Example

```
<!doctype html>
<html>
    <head>
<title>Float positioning</title>
<style type = 'text/css'>
#col1 {
    float: left;
    width: 150px;
    background-color: lightskyblue;
}
#col2 {
    float: left;
    width: 120px;
    background-color: yellow;
}
</style>
</head>

<body>
<h1>Float positioning</h1>
<p id = "col1">Lorem ipsum . . .
<p id = "col2">Lorem ipsum . . .
<p id = "col3">Lorem ipsum . . .
</body>
</html>
```

Float Positing Example



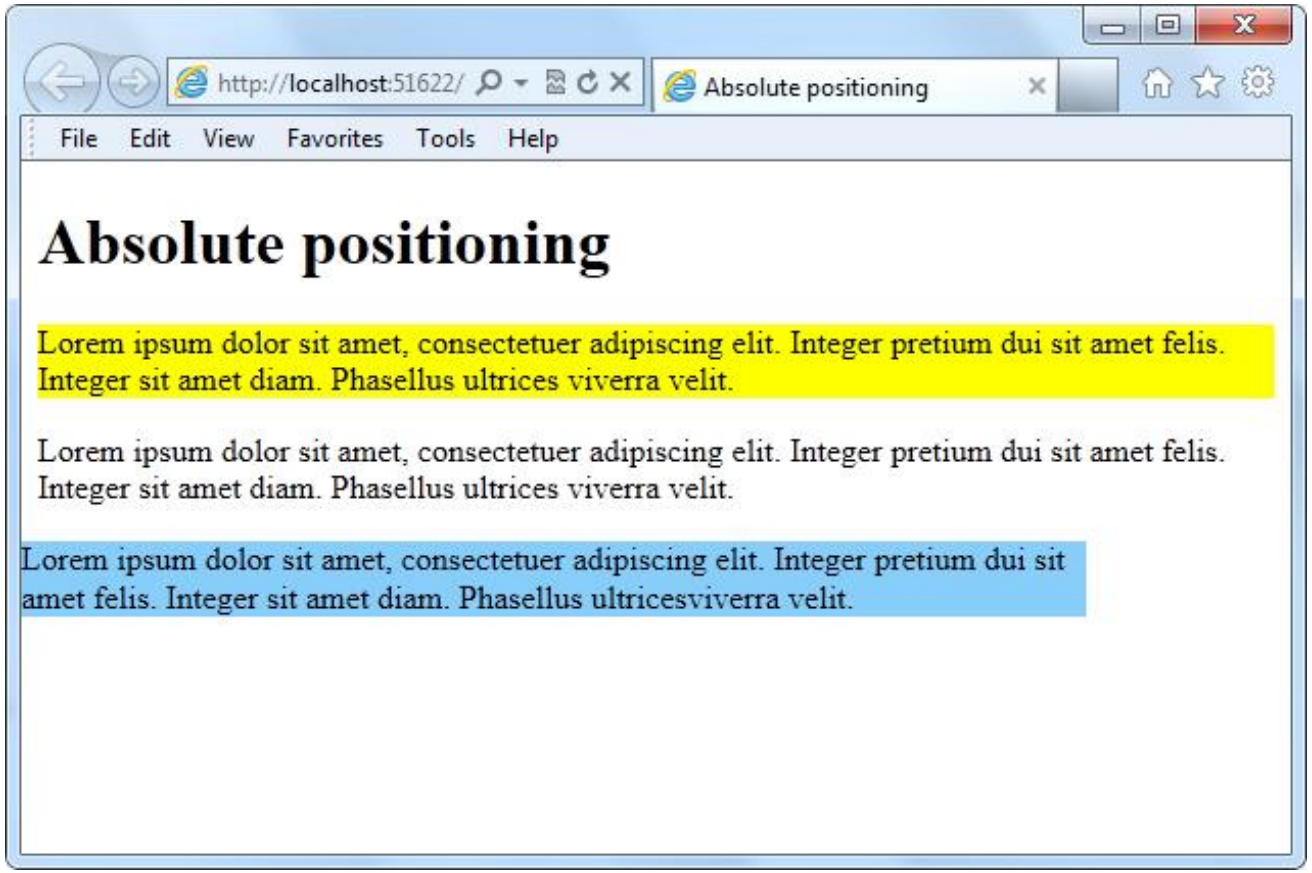
Absolute Positioning Example

```
<!doctype html>
<html>
<head>
<title>Absolute positioning</title>
<style type = 'text/css'>
#col1 {
    position: absolute;
    bottom: 100px;
    right:  100px;
    background-color: lightskyblue;
}
#col2 {
    background-color: yellow;
}

</style>

</head>
<body>
<h1>Absolute positioning</h1>
<p id = "col1">Lorem ipsum . . .
<p id = "col2">Lorem ipsum . . .
<p id = "col3">Lorem ipsum . . .
</body>
</html>
```

Absolute Positing Example



Bounding Box

- A **bounding box** is a rectangular border around content -- text, an image, or a shape -- that enables you to move, rotate, or scale the content of the box.
- Bounding box can be visible or invisible.

Overflow

- When an element overflows its bounding box, and its overflow is set to scroll, all the content of the element stays within the box; none of the overflow appears outside the box. This is referred to as ***scrolling overflow.***
- ***Visible overflow*** writes over the content that follows it.
- ***Hidden overflow*** makes overflow invisible.

Overflow

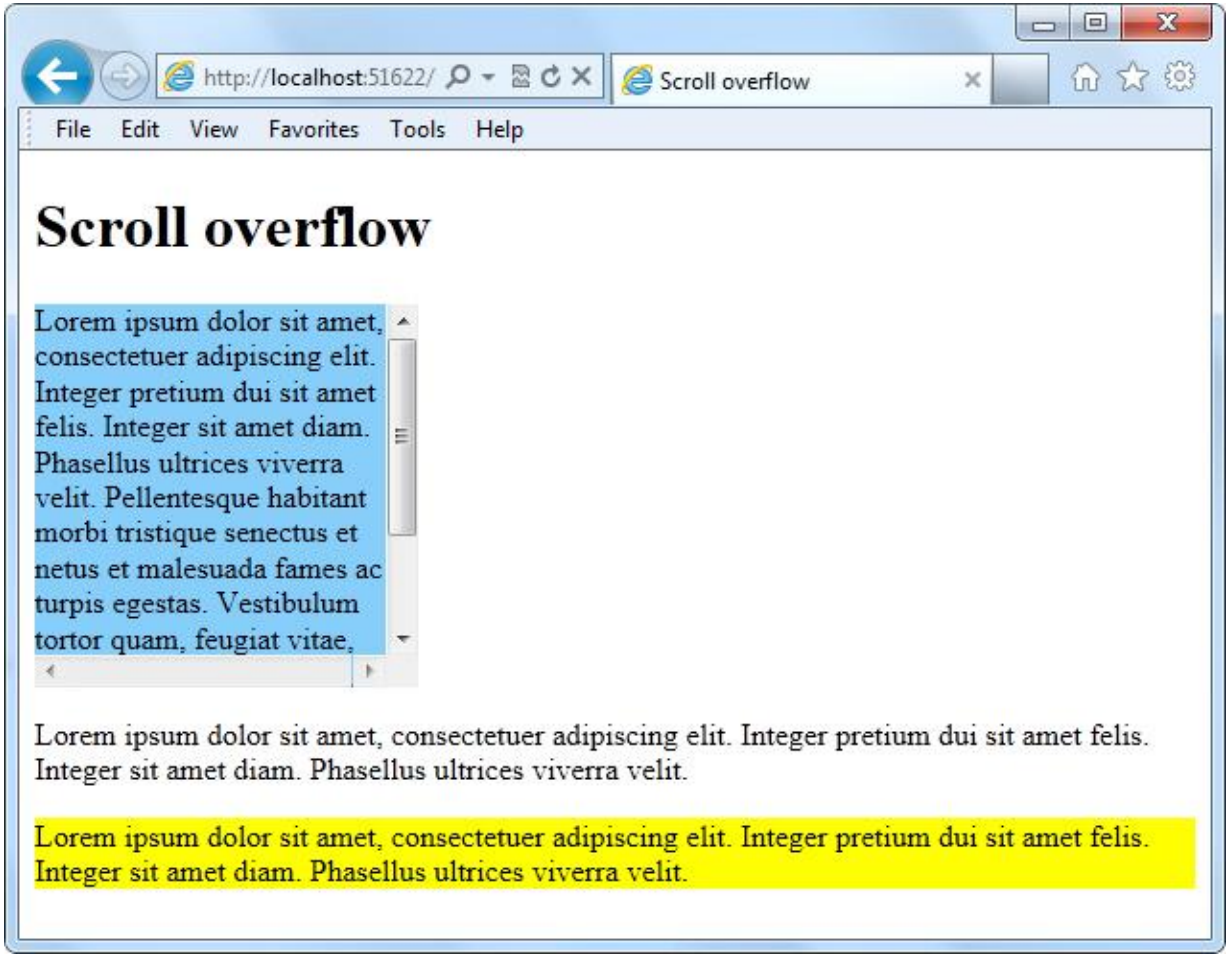
- `overflow` property
- Values:
  - `Scroll`
  - `Visible`
  - `Hidden`

Scrolling Overflow Example

```
<!doctype html>
<html>
<head>
<title>Scroll overflow</title>
<style type = "text/css" >
#col1 {
width: 200px;
height: 200px;
background-color: lightskyblue;
overflow: scroll;
}
#col3 {
background-color: yellow;
}
</style>
</head>

<body>
<h1>Scroll overflow</h1>
<p id = "col1">Lorem ipsum . . .</p>
<p id = "col2">Lorem ipsum . . .</p>
<p id = "col3">Lorem ipsum . . .</p>
</body>
</html>
```

Scrolling Overflow Example

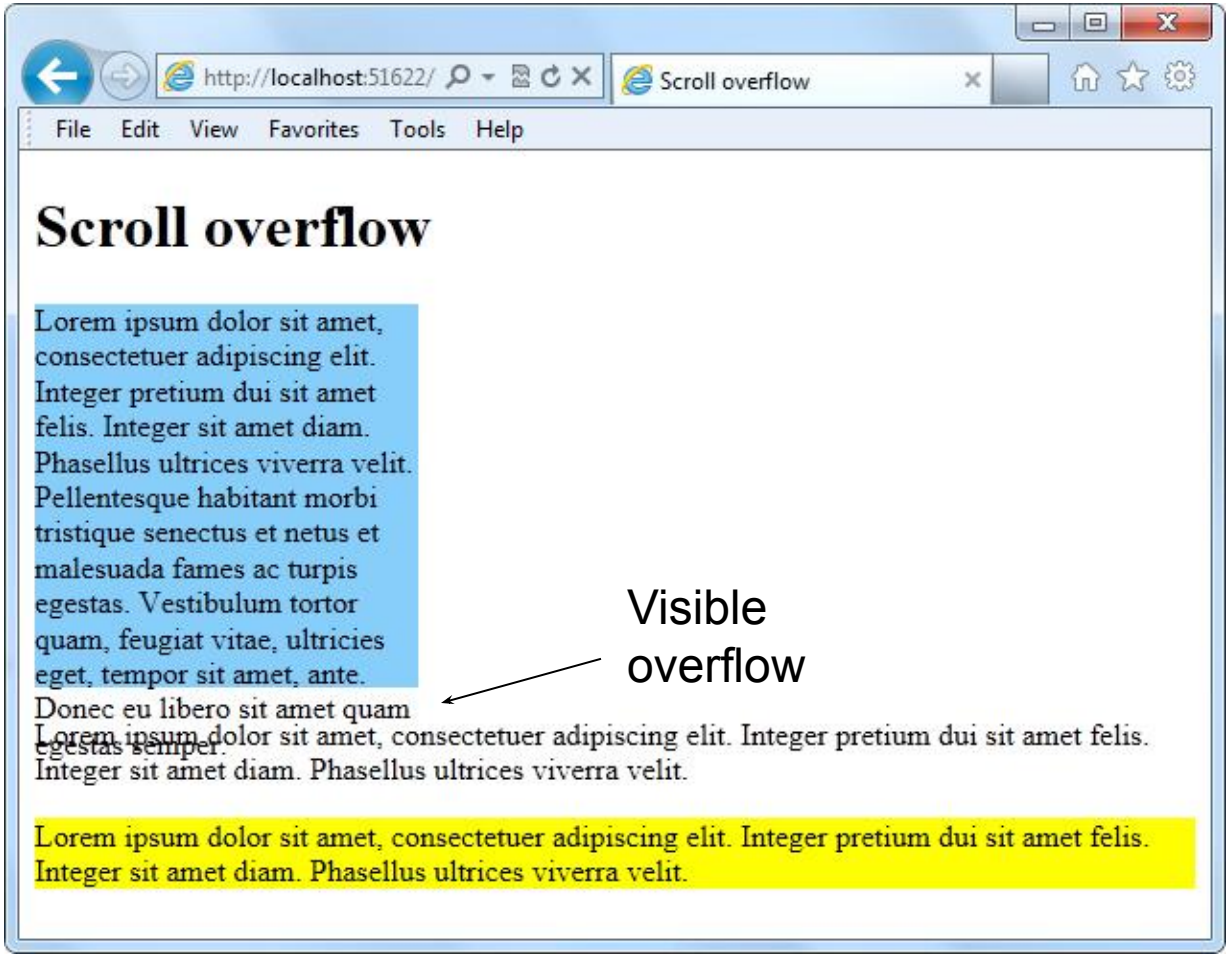


Visible Overflow Example

```
<!doctype html>
<html>
<head>
<title>Scroll overflow</title>
<style type = "text/css" >
#col1 {
width: 200px;
height: 200px;
background-color: lightskyblue;
overflow: visible;
}
#col3 {
background-color: yellow;
}
</style>
</head>

<body>
<h1>Scroll overflow</h1>
<p id = "col1">Lorem ipsum . . .</p>
<p id = "col2">Lorem ipsum . . .</p>
<p id = "col3">Lorem ipsum . . .</p>
</body>
</html>
```

Visible Overflow Example

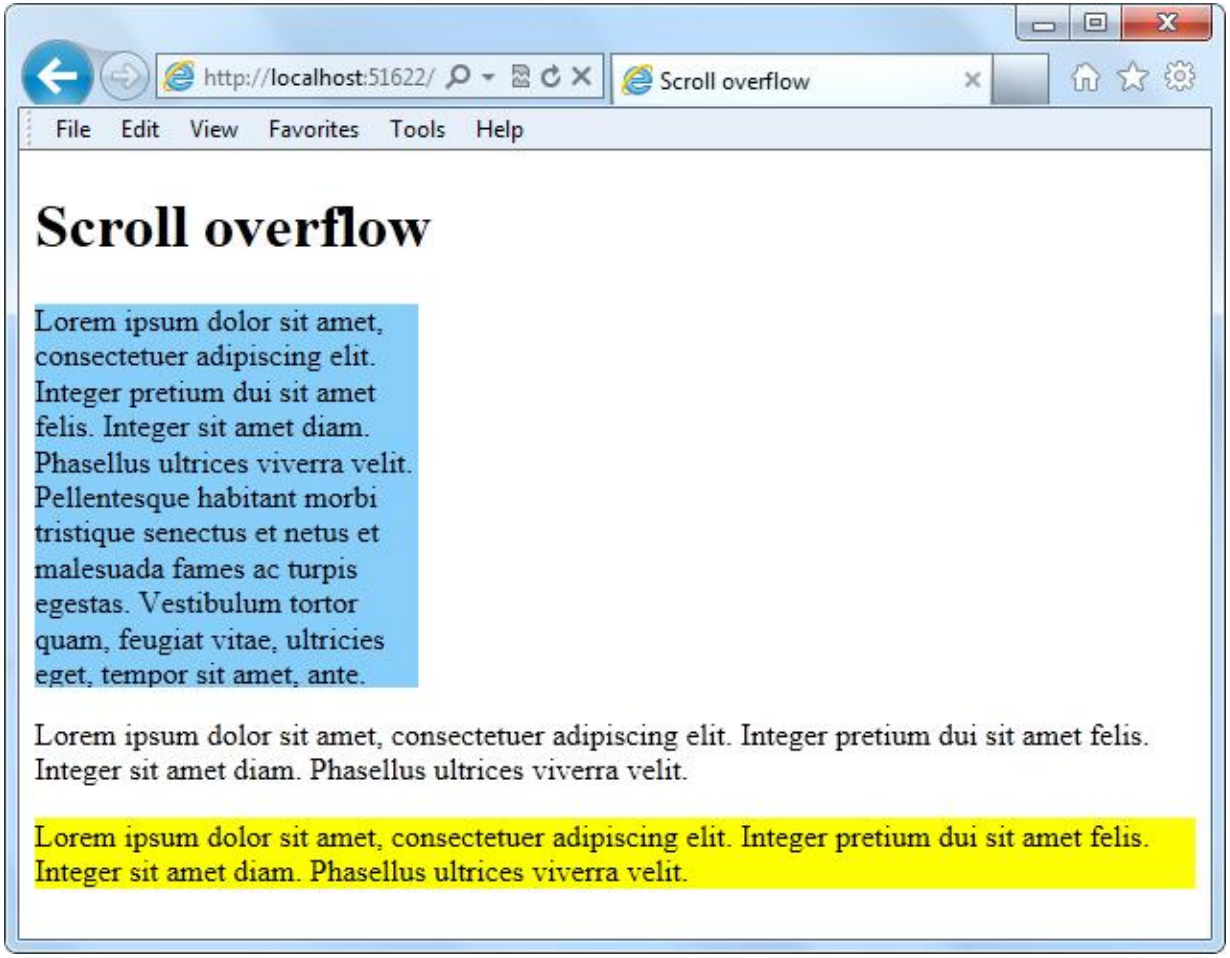


Hidden Overflow Example

```
<!doctype html>
<html>
<head>
<title>Scroll overflow</title>
<style type = "text/css" >
#col1 {
width: 200px;
height: 200px;
background-color: lightskyblue;
overflow: hidden;
}
#col3 {
background-color: yellow;
}
</style>
</head>

<body>
<h1>Scroll overflow</h1>
<p id = "col1">Lorem ipsum . . .</p>
<p id = "col2">Lorem ipsum . . .</p>
<p id = "col3">Lorem ipsum . . .</p>
</body>
</html>
```

Hidden Overflow Example



Practice Task

Contacts

Europe Headquarters

52 V. Velykoho Str.
Lviv 79053, Ukraine

Tel: +380-32-240-9090
Fax: +380-32-240-9080

E-mail: info@softserveinc.com
Website: **www.softserveinc.com**

US Headquarters

12800 University Drive, Suite 250
Fort Myers, FL 33907, USA

Tel: 239-690-3111
Fax: 239-690-3116

Thank You!