TDDD97 - Web Programming

Client-side development

Sahand Sadjadee

Dept. of Computer and Information Science

Linköping University

Outline

- Why web apps?
- The Front-end
  - HTML
  - CSS
  - Javascript
- HTML5
- Selenium Testing
- Responsive Web Design(RWD)

Why do we **develop** web applications?

Desktop application

Mr Gates

Outlook 2003 released

Users buy and install

Outlook 2007 released

New marketing and shipping of products to stores required.

Mr Gates

Users keep the old Office package, feel no need to upgrade or does not even know it can be upgraded.

Why do we **develop** web applications?

Desktop application

Mr Gates

Outlook 2003 released

Users buy and install

Outlook 2007 released

New marketing and shipping of products to stores required.

Mr Gates

Users keep the old Office package, feel no need to upgrade or does not even know it can be upgraded.

Web application

Office 365 released

Mr Gates

Users create accounts on the web and pay a small sum every month.

New updates to the web page and application.

The next time the user logs in the new changes will appear.

Mr Gates

Users get immediate access to new updates without marketing and shipping to stores.

Continuous income every month, as long as the customers are happy.

Why do we **use** web applications?

- No installation required (if the computer has a web browser).

- The underlying platform does not matter, as long as you have a web browser and internet access.

- The same documents/files available from all computers you log in from. Less, but not none, dependent on local storage comparing to Desktop and even mobile applications.

- You can login from any location in the world and for example read your e-mail.

One of the most popular features of Facebook is that people can use it during vacation and at work, for example.

Imagine that Facebook was a desktop application that required installation. Would the company be valued at this level?
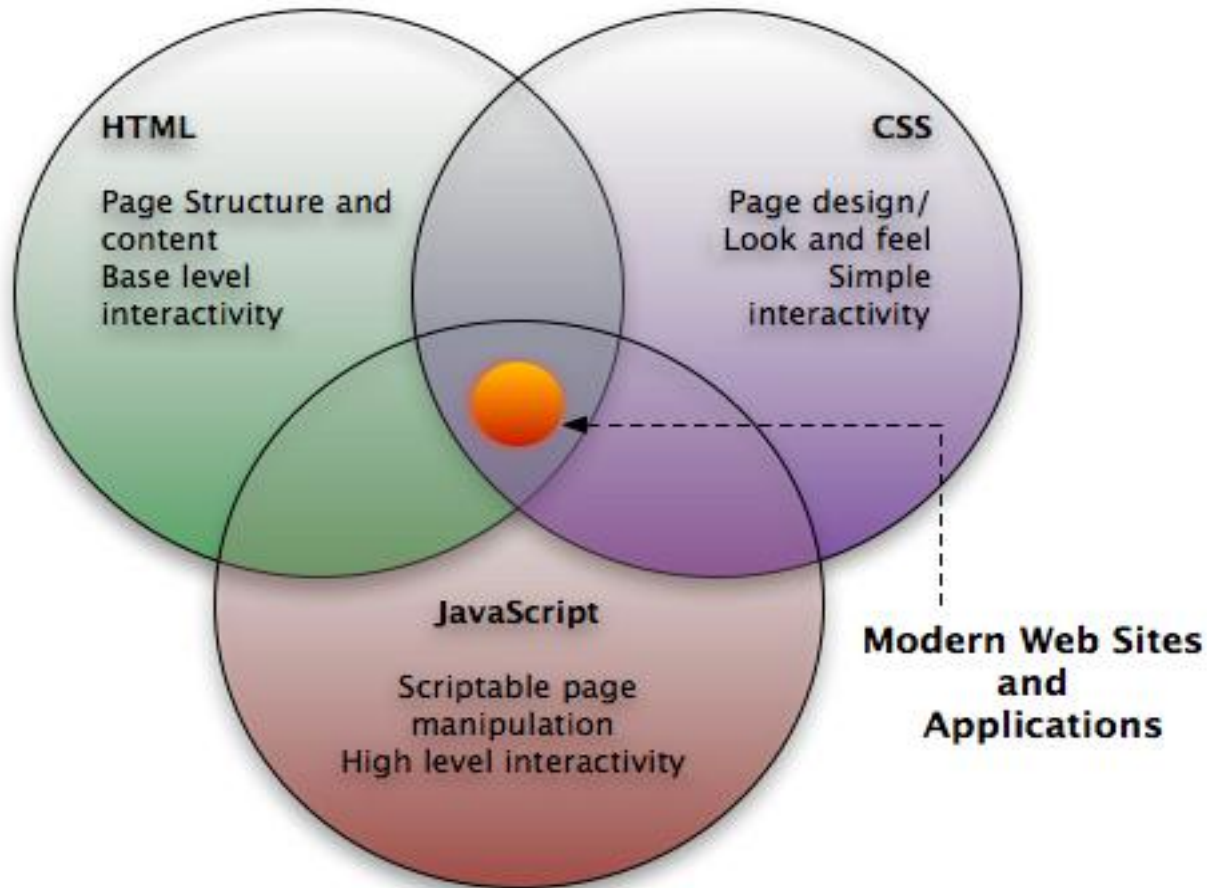
The Front-end
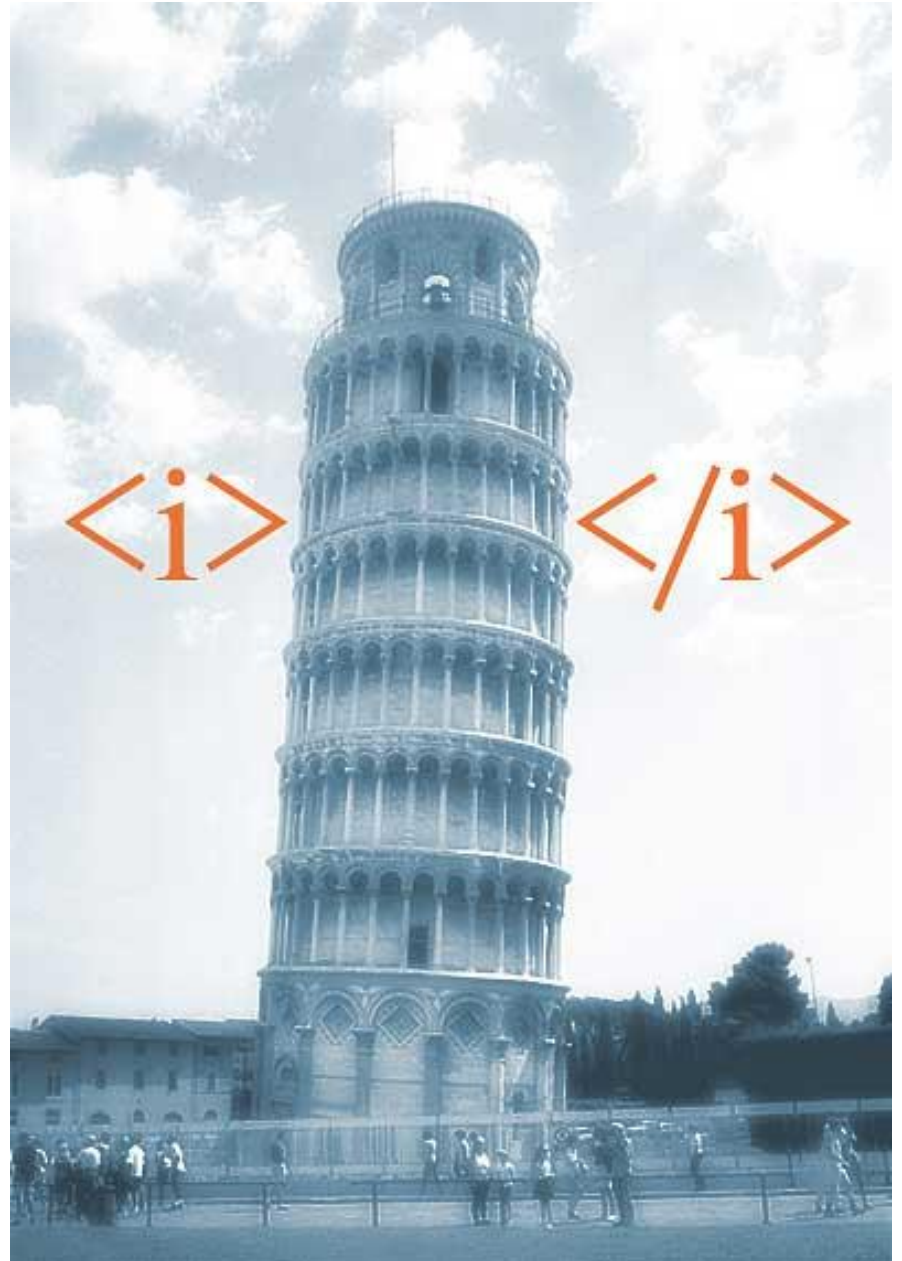
Front-end in web applications

- Web apps like other types of applications need front-end, Graphical User Interface plus the logic related to it, to be available for the users to communicate with.
- The front-end in case of web applications is sent first to the client and then executed by and inside of a web browser located at the client-side.
- In case of SPA, the front-end is mostly delivered to the client in the beginning.
- In case of MPA, the front-end is sent every time a page is requested.

The building blocks

HTML
Page Structure and content
Base level interactivity

CSS
Page design/ Look and feel
Simple interactivity

JavaScript
Scriptable page manipulation
High level interactivity

Modern Web Sites and Applications

http://www.99points.info/2010/03/50-css-techniques-and-tips-which-you-always-need/

HyperText Mark-up Language

HTML

Server

1. Get a pure text document, which the user can read with some difficulty.

Chapter 1 Lorem ipsum dolor sit amet, consectetur adipiscing elit. Avsnitt 1 Vestibulum venenatis, velit sed cursus ultricies, sem nisi fringilla erat.

Server

2. Get an HTML document, which is even more difficult to read.

<h1>Chapter 1</h1>
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. </p>
<h2>Section 1</h2>
<p>Vestibulum venenatis, velit sed cursus ultricies, sem nisi fringilla erat.</p>

1. Get a pure text document, which the user can read with some difficulty.

Chapter 1 Lorem ipsum dolor sit amet, consectetur adipiscing elit. Avsnitt 1 Vestibulum venenatis, velit sed cursus ultricies, sem nisi fringilla erat.

Server

2. *Get an HTML document, which is even more difficult to read.*

```
<h1>Chapter 1</h1>
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. </p>
<h2>Section 1</h2>
<p>Vestibulum venenatis, velit sed cursus ultricies, sem nisi fringilla erat.</p>
```

1. *Get a pure text document, which the user can read with some difficulty.*

Chapter 1 Lorem ipsum dolor sit amet, consectetur adipiscing elit. Avsnitt 1 Vestibulum venenatis, velit sed cursus ultricies, sem nisi fringilla erat.

3. *If we let the web browser handle the HTML document, it will show it as a readable and nicely formatted document.*

Chapter 1

Lorem ipsum dolor sit amet,   consectetur adipiscing elit.

Section 1

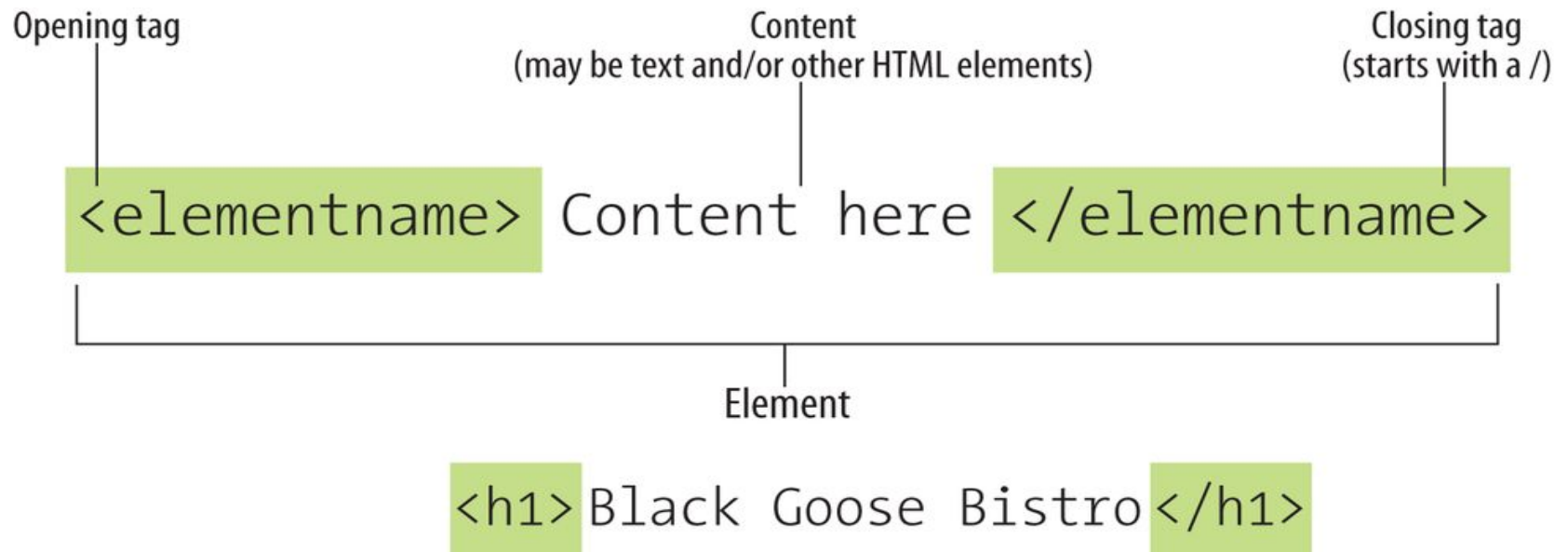Vestibulum venenatis, velit sed cursus ultricies, sem nisi fringilla erat.

HTML

HTML is the standard markup language for creating Web pages.

- HTML stands for Hyper Text Markup Language
- HTML describes the structure of Web pages using markup
- **HTML elements** are the building blocks of HTML pages
- HTML elements are represented by tags
- HTML tags label pieces of content such as "heading", "paragraph", "table", and so on
- Browsers do not display the HTML tags, but use them to render the content of the page
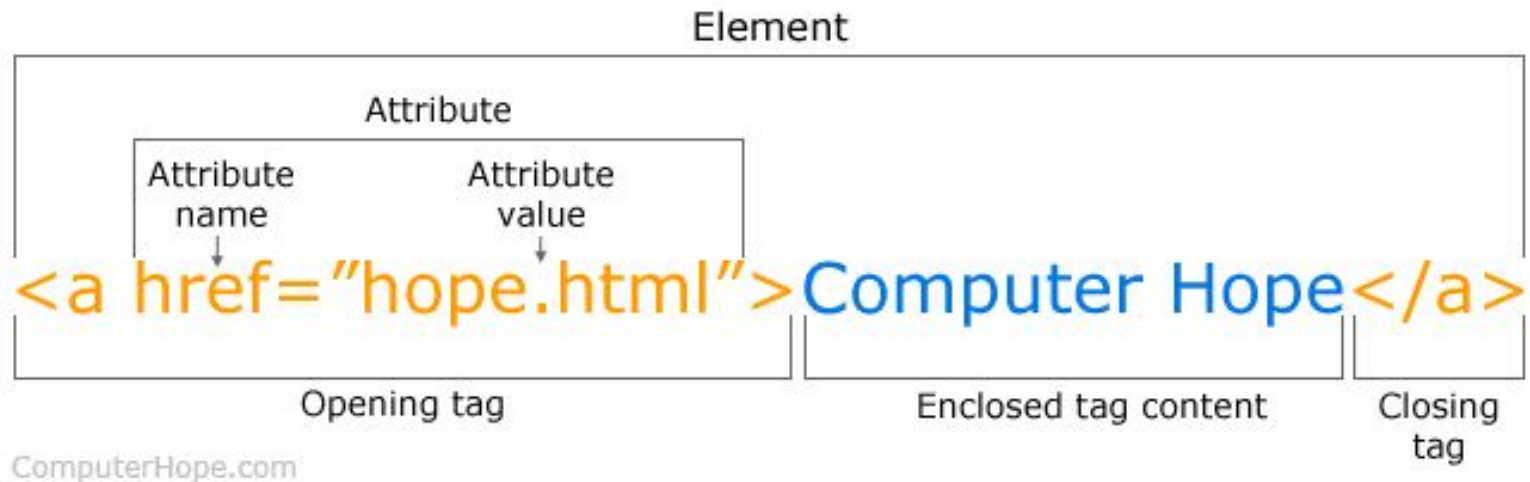
HTML Elements

HTML elements are the building blocks of HTML pages. With HTML constructs, images and other objects, such as interactive forms, may be embedded into the rendered page. It provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items.

Opening tag

Content
(may be text and/or other HTML elements)

Closing tag
(starts with a /)

`<elementname>` Content here `</elementname>`

Element

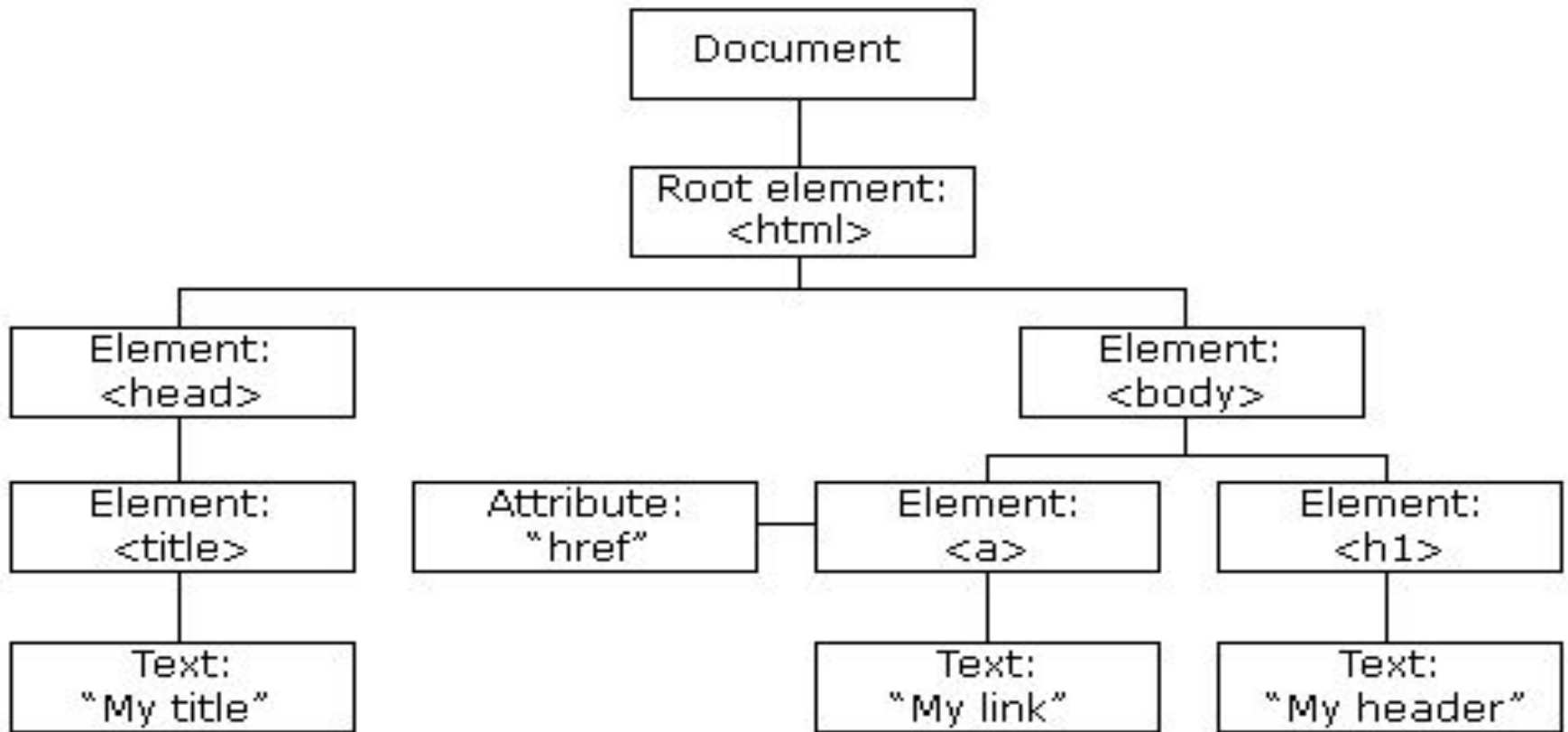`<h1>`Black Goose Bistro`</h1>`

HTML Attributes

- Attributes provide **additional information** about an element which may affect how the element is rendered or behave.
- Attributes are always specified in **the start tag**
- Attributes usually come in name/value pairs like: **name="value"**

Element

Attribute

Attribute name

Attribute value

`<a href="hope.html">Computer Hope</a>`

Opening tag

Enclosed tag content

Closing tag

ComputerHope.com

HTML Content Structure

https://www.w3schools.com/js/js_htmldom_navigation.asp

*) Usually, an element can contain other elements.

HTML Tags, example

HTML	Rendering example
`<h1>Heading</h1>`	# Heading
`<h2>Heading</h2>`	## Heading
`<p>Paragraph</p>`	Paragraph

`<ul>`
`   <li>Item 1</li>`
`   <li>Item 2</li>`
`</ul>`
	• Item 1
	• Item 2

`<table>`
`   <tr>`
`      <td>Cell 1</td>`
`      <td>Cell 2</td>`
`   </tr>`
`   <tr>`
`       <td>Cell 3</td>`
`       <td>Cell 4</td>`
`   </tr>`
`</table>`
	<table><tr><td>Cell 1</td><td>Cell 2</td></tr><tr><td>Cell 3</td><td>Cell 4</td></tr></table>

`<a href="http://www.google.com">Google</a>`	Google (This is an example of an HTML attribute, in form of the "href" hypertext reference)

HTML elements reference: https://developer.mozilla.org/en-US/docs/Web/HTML/Element

Demo

Try some HTML….

https://www.w3schools.com/html/tryit.asp?filename=tryhtml_default

There are many web browsers on the market, and they are not all alike.

A web application may look and behave slightly differently in different web browsers. It's very crucial to test your implementation in the major web browsers before making it available to the public.

HTML – Class and ID

class

*Marks that an element is of a certain type. HTML documents may contain **several elements** with the same class value.*

```
<div class="ingress">Detta är text</div>

<a class="external" href="...">External</a>

<button class="image-button"></button>
```

id

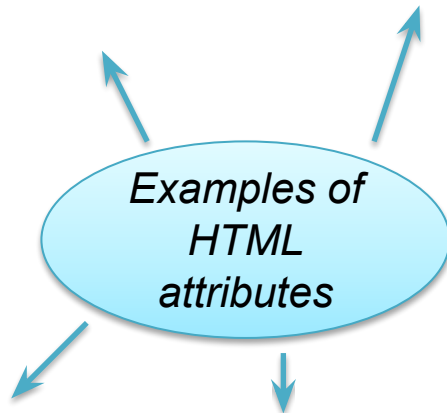*Marks a unique element in the HTML code. The HTML document may only contain **one element** with a certain id value.*

```
<button id="sendEmail">Send</button>

<a id="googleLink" href="...">Google</a>

<div id="leftContent">....</div>
```

Elements can have both class and id values defined.

HTML Form

```
<form method="POST" action="">
  Name: <input type="text" /><br />
  <input type="radio" name="gender" value="female" /> Female<br />
  <input type="radio" name="gender" value="male" /> Male<br />
  <button type="submit">Send</button>
</form>
```
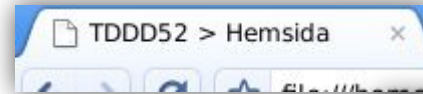
Name: [   ]
○ Female
○ Male
[Send]

Examples of HTML attributes

```
<form method="POST" action="">
  <input type="checkbox" value="CC" />Send CC<br/>
  Message: <textarea rows="3"></textarea><br />
  <button type="submit">Send</button>
</form>
```

☐ Send CC

Message: [   ]
[Send]

HTML – Head unique tags

```
<html>
    <head>
        <title>TDDD52 &gt; Hemsida</title>
    </head>
    <body>
    </body>
</html>
```

```
<html>
    <head>
        <title> Amazon.com: Online Shopping for Electronics, Apparel, Computers, DVDs &amp; more</title>

        <meta name="description" content="Online retailer of books, movies, music and games along with
electronics, toys, apparel, sports, tools, groceries and general home and garden         items." />
    </head>
    <body>
    </body>
</html>
```

Special characters – HTML Entities

- If you use the "<" or ">" character in your text, the web browser will interpret this as the beginning or end of a tag, and everything will be messed up.

  Instead use: < or >

- If you would like to use a character not available on the keyboard, for example the copyright sign (©):

  Use: ©

- Search the web of a complete list of special characters…

Validation of HTML code

- http://validator.w3.org
- Correctly written XHTML code removes the guesswork that the web browser otherwise would have to preform.
- You website will have a better chance of looking as intended in all web browsers.
- If you get errors, correct them one at a time until you get a green bar.

HTML

Partitioning and semantics

- *Logical partitioning of the text in a document.*

- *Gives pieces of text a semantic meaning, such as headings.*

- *Allows the web browser to interpret the document before showing it.*

- *Allows search engines to better understand the content of a web site.*

Forms

- *Provides document components for forms.*

- *Buttons, text boxes, radio buttons, check boxes etc.*

- *These forms are an important part of the communication between the user and the web server.*

- *Examples: credit-card transactions, comment fields, etc.*

Links

- *Links documents to other documents.*

- *Creates a network of documents that the user views as a "site".*

- *The links can give hints to search engines about content and relevance.*

HTML, XHTML, HTML5

Yesterday

- HTML 4 – Hyper Text Mark-up Language

- XHTML - Extensible Hyper Text Mark-up Language

- Two different specifications for marking up text using tags

Today

- HTML5 – The future of HTML

- HTML5 is not only for semantics, forms, and links, but also for advanced techniques such as local databases, video, sockets, etc.

- Currently, we are at the beginning of this development, and all HTML5 features are not implemented in all web browsers. You can measure how much your web browser supports HTML5 by using the following link:

  https://html5test.com/

Cascading Style Sheet

CSS
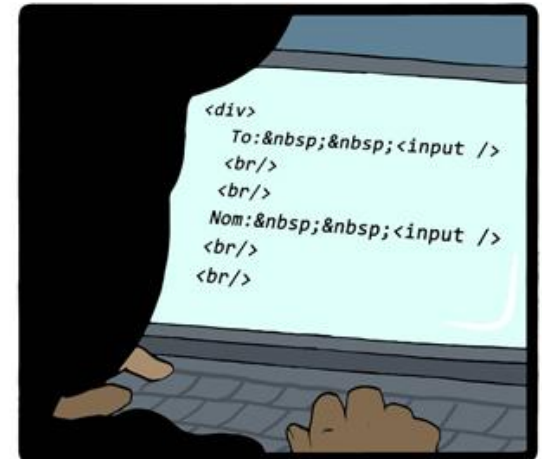
CommitStrip.com

CSS

- Cascading Style Sheet

- Presentation of content

- *Cascade* can be interpreted as "**inherit and overwrite**"

- An important and powerful tool to make web sites usable.
- Can change colors, sizes, placements, etc.
- Interpreted by web browsers, which can lead to somewhat different results.

CSS – Selector

A selector is a syntax element that determines which elements should be affected by which rules.

Selector	Explanation
h1	A certain type of element, such as all *h1* elements
.ingress	All elements with attribute "class" set to a certain value, such as "class=ingress"
#sendButton	An element with a certain id, such as "id=sendButton"
div.ingress	All *div* elements with "class=ingress"
div h1	*Nested* selectors, such as all *h1* within *div*
div.ingress h1	All combinations work

CSS – Properties

A property determines what should be changed and value determines what it should be changed to.

Property	Values
color	RGB color code, sets the text in the selected element to the new color.
width	Sets the with of an element, uses units such as px, %, etc.
border	Value could be "5px solid red", which gives a 5px thick solid border.
border-style	Sets the border to a different type, such as dotted, dashed, double, groove, ridge, solid, etc.
	There are lots of different *property-value* you can use.

CSS – Examples

CSS	Result

a {
  color: #F00;
}
	All links become red instead of default blue.

div h1 {
   font-size: 12px;
   border: 1px solid #CCC;
}
	All h1 tags nested under div tags get smaller font and grey border.

div.ingress {
 font-weight: bold;
}
	All div tags that have "class=ingress" get bold font

img {
  cursor: pointer;
}
	All images change the cursor to a *hand* instead of the default cursor when you hover the mouse over the image.

CSS – Hover selector

```
a:hover {
    background-color: yellow;
}
```

A selector that first applies its rules when the mouse is hovered over the element.

www.google.com   *www.google.com*

Check out the following link if you use :hover on you web sites: http://w3schools.com/cssref/sel_hover.asp There are some things to keep in mind when you use it in combination with the a element.

HTML – span - inline

\<span\>

Use when you want to mark something in a text flow, such as **\<span class="red"\>this should be in red\</span\>** *but when the text otherwise should flow normally...*

↓

Use when you want to mark something in a text flow, such as **this should be in red** *but when the text otherwise should flow normally...*

Used to group inline elements in a document.

This group can then be modified with CSS or JavaScript.

HTML – div - block

<div>

When you want to group bock elements
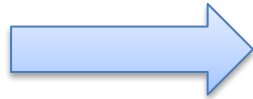*<div class="green-border">*
   <ul>
      <li>Item 1</li1>
      <li>Item 2</li1>
      <li>Item 3</li1>
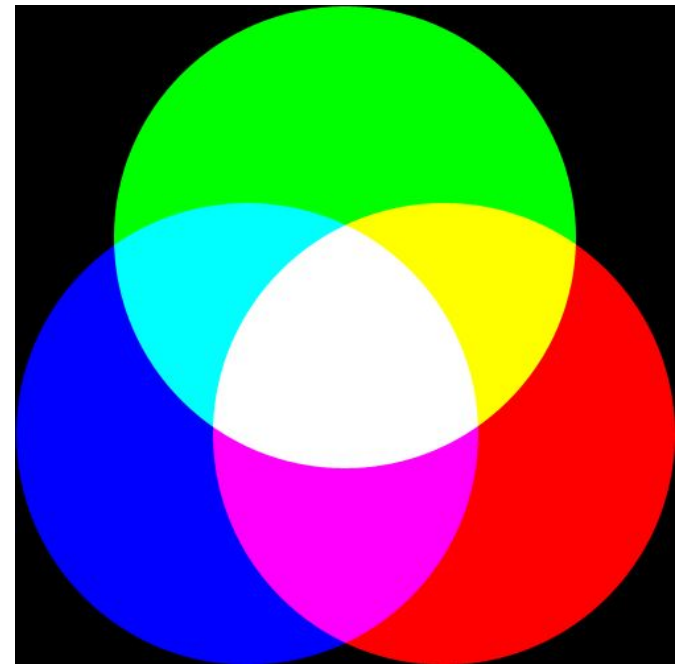   *<li>Item 4</li1>*
   *
*
</div>

- *Item 1*
- *Item 2*
- *Item 3*
- *Item 4*

Colors

- Additive color is used on the web

- We mix the colors red, green, and blue (RGB)

- Hexadecimal: 0 1 2 3 4 5 6 7 8 9 A B C D E F

- #000000 or #000 = Black
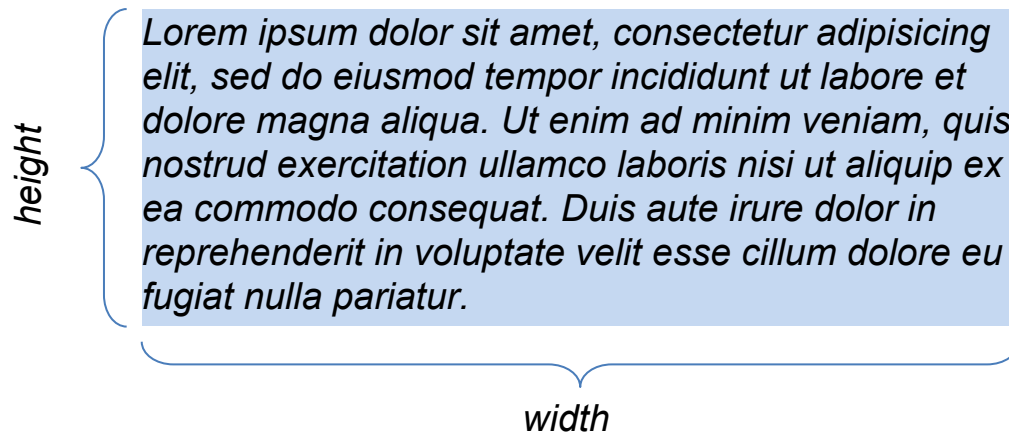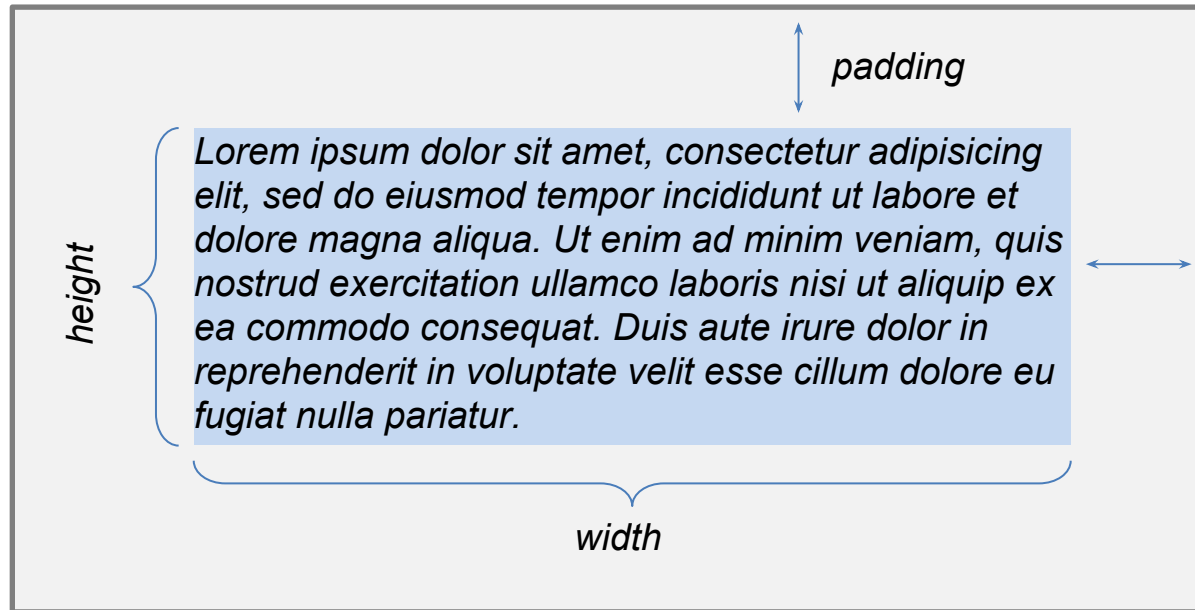- #FFFFFF or #FFF = White
- #FF0000 or #F00 = Red

Colors

- Use a list or a program to look up the color you want

Box model

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur.

height

width

Box model

Box model

border

padding

height

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur.

width

Box model

border

padding

height

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur.

width

margin

The actual width is = width + padding + border + margin

CSS – Float & Clear

DIV

DIV

DIV

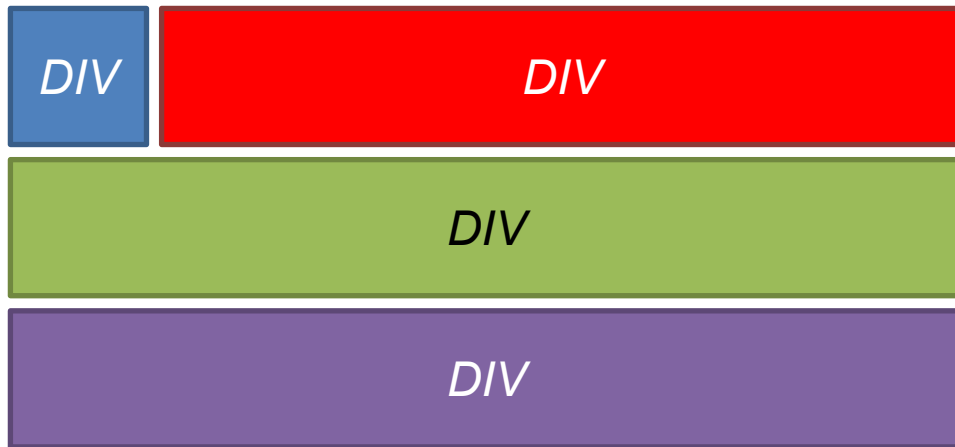| DIV |
|:---:|

Suppose we have four consecutive div tags.

When the web browser sees the next div tag, it will make a line break and then render the whole div tag.
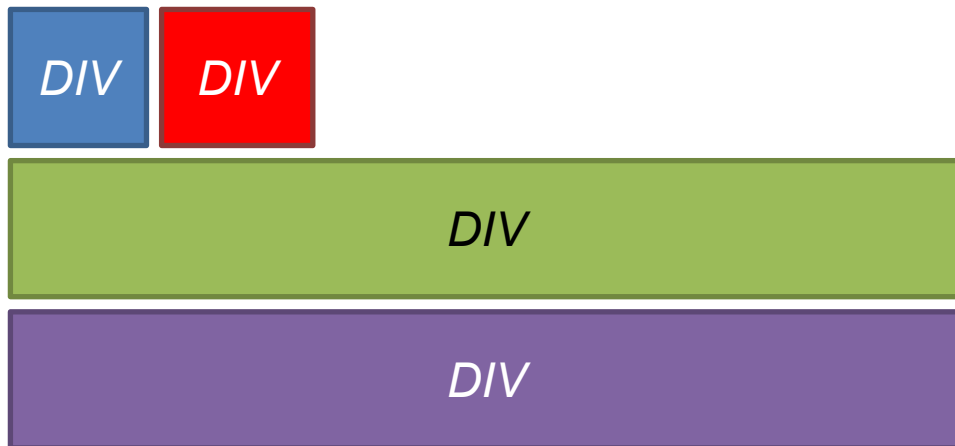
CSS – Float & Clear

If we set the blue div tag to **float: left;** it will be placed to the far left. The next element will be placed directly after.

The red div tag does not have float, and the green one will be placed on the next line.

CSS – Float & Clear

*If we also set the red tag to* **float: left;** *the green one will come directly after.*

If we want to avoid that the next element is placed directly after the red one, we can set **clear: left;** *for the green element.*

CSS – Float & Clear

- You can also use *float: right;* and *clear: right;*

- Setting float for elements makes it possible to create other layouts than the default one.

Exempel 1 - Navigering på sidan

Sidhuvud	
Navigering	Sidans innehåll
Sidfot	
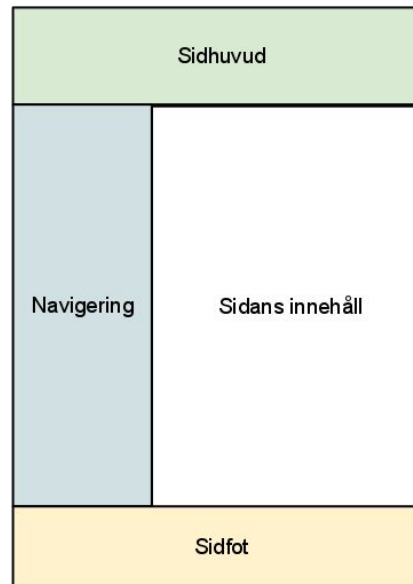
Exempel 2 - Navigering ovan

Sidhuvud
Navigering
Sidans innehåll
Sidfot
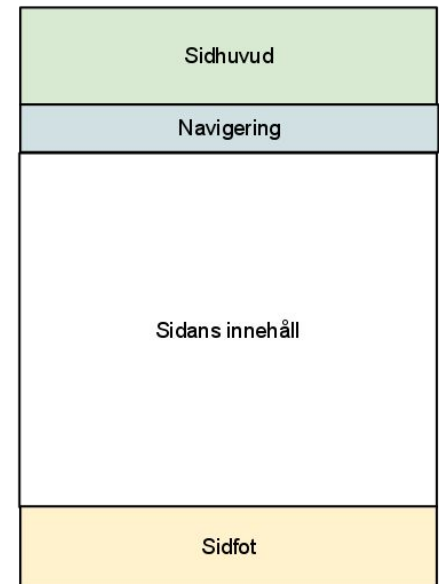
Using float enables us to place elements in other ways than plain HTML.

CSS Validation

- http://jigsaw.w3.org/css-validator/

- Just as for HTML, correct CSS is necessary for removing the guesswork for the web browser.

- If you write large style sheets it is a good idea to validate often.

- A "manual" validation is still necessary to ensure that the website looks as intended in the major web browsers.
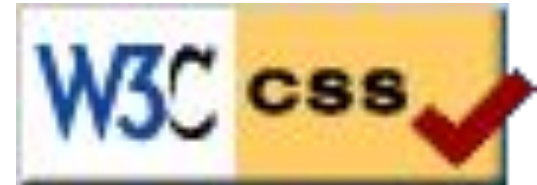
Congratulations! No Error Found.

Separation of content and presentation

- Easier to create, maintain, and reuse.

- This separation is essential when developing dynamic web pages.

- It is possible to create different different designs depending on the device and its resolution.

HTML

CSS CSS CSS
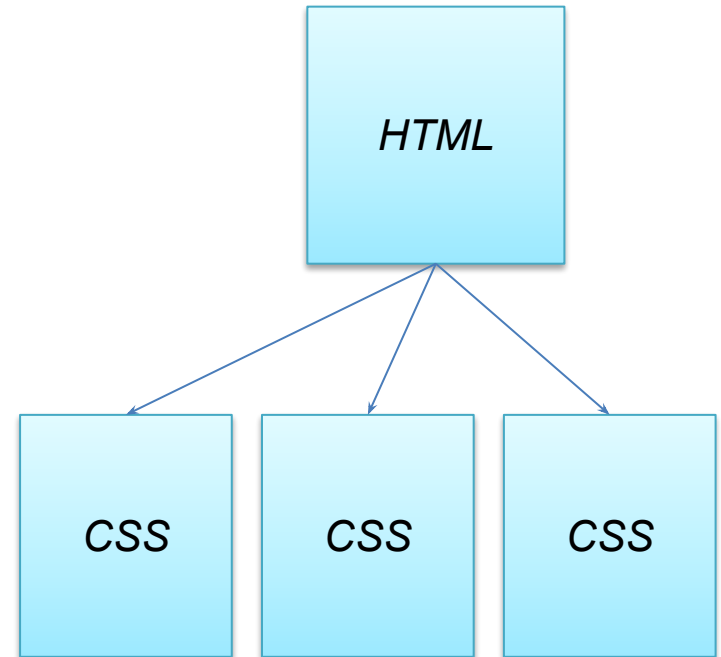
Some example pages…

Zen Garden

The Beauty of CSS Design

A demonstration of what can be accomplished visually through CSS-based design. Select any style sheet from the list to load it into this page.

Download the sample html file *and* css file

The Road to Enlightenment

Littering a dark and dreary road lay the past relics of browser-specific tags, incompatible DOMs, and broken CSS support.

Today, we must clear the mind of past practices. Web enlightenment has been achieved thanks to the tireless efforts of folk like the W3C, WaSP and the major browser creators.

The css Zen Garden invites you to relax and meditate on the important lessons of the masters. Begin to see with clarity. Learn to use the (yet to be) time-honored techniques in new and invigorating fashion. Become one with the web.

So What is This About?

There is clearly a need for CSS to be taken seriously by graphic artists. The Zen Garden aims to excite, inspire, and encourage participation. To begin, view some of the existing designs in the list. Clicking on any one will load the style sheet into this very page. The code remains the same, the only thing that has changed is the external .css file. Yes, really.

CSS allows complete and total control over the style of a hypertext document. The only way this can be illustrated in a way that gets people excited is by demonstrating what it can truly be, once the reins are placed in the hands of those able to create beauty from structure. To date, most examples of neat tricks and hacks have been demonstrated by structurists and coders. Designers have yet to make their mark. This needs to change.

Participation

Graphic artists only please. You are modifying this page, so strong CSS skills are necessary, but the example files are commented well enough that even CSS novices can use them as starting points. Please see the CSS Resource Guide for advanced tutorials and tips on working with CSS.

You may modify the style sheet in any way you wish, but not the HTML. This may seem daunting at first if you've never worked this way before, but follow the listed links to learn more, and use the sample files as a guide.

Download the sample html file and css file to work on a copy locally. Once you have completed your masterpiece (and please, don't submit half-finished work) upload your .css file to a web server under your control. Send us a link to the file and if we choose to use it, we will spider the associated images. Final submissions will be placed on our server.

Benefits

Why participate? For recognition, inspiration, and a resource we can all refer to when making the case for CSS-based design. This is sorely needed, even today. More and more major sites are taking the leap, but not enough have. One day this gallery will be a historical curiosity; that day is not today.

Requirements

We would like to see as much CSS1 as possible. CSS2 should be limited to widely-supported elements only. The css Zen Garden is about functional, practical CSS and not the latest bleeding-edge tricks viewable by 2% of the browsing public. The only real requirement we have is that your CSS validates.

Unfortunately, designing this way highlights the flaws in the various implementations of CSS. Different browsers display differently, even completely valid CSS at times, and this becomes maddening when a fix for one leads to breakage in another. View the Resources page for information on some of the fixes available. Full browser compliance is still sometimes a pipe dream, and we do not expect you to come up with pixel-perfect code across every platform. But do test in as many as you can. If your design doesn't work in at least IE5+/Win and Mozilla (run by over 90% of the population), chances are we won't accept it.

We ask that you submit original artwork. Please respect copyright laws. Please keep objectionable material to a minimum; tasteful nudity is acceptable, outright pornography will be

C.S.S. ZEN GARDEN

A demonstration of what can be accomplished visually through CSS-based design. Select any style sheet from the list to load it into this page.

DOWNLOAD THE SAMPLE HTML FILE AND CSS FILE

THE ROAD TO ENLIGHTENMENT

Littering a dark and dreary road lay the past relics of browser-specific tags, incompatible DOMs, and broken CSS support.

Today, we must clear the mind of past practices. Web enlightenment has been achieved thanks to the tireless efforts of folk like the W3C, WaSP and the major browser creators.

The css Zen Garden invites you to relax and meditate on the important lessons of the masters. Begin to see with clarity. Learn to use the (yet to be) time-honored techniques in new and invigorating fashion. Become one with the web.

SO WHAT IS THIS ABOUT?

There is clearly a need for CSS to be taken seriously by graphic artists. The Zen Garden aims to excite, inspire, and encourage participation. To begin, view some of the existing designs in the list. Clicking on any one will load the style sheet into this very page. The code remains the same, the only thing that has changed is the external .css file. Yes, really.

CSS allows complete and total control over the style of a hypertext document. The only way this

The Beauty of CSS Design

Fig. A

SELECT A DESIGN

Under the Sea!
by Eric Stoltz

Make 'em Proud
by Michael McAghon and Scotty Reifsnyder

Orchid Beauty
by Kevin Addison

Oceanscape
by Justin Gray

CSS Co., Ltd.
by Benjamin Klemm

Sakura
by Tatsuya Uchida

Kyoto Forest
by John Politowski

A Walk in the Garden
by Simon Van Hauwermeiren

ARCHIVES

Fig. B

ICTINEO-MONTURIOL

GET STARTED

Download the sample **html file** and **css file**

THE PATH TO ACHIEVEMENT

Littering a dark and dreary road lay the past relics of browser-specific tags, incompatible DOMs, and broken CSS support.

Today, we must clear the mind of past practices. Web enlightenment has been achieved thanks to the tireless efforts of folk like the W3C, WaSP and the major browser creators.

The css Zen Garden invites you to relax and meditate on the important lessons of the masters. Begin to see with clarity. Learn to use the (yet to be) time-honored techniques in new and invigorating fashion. Become one with the web.

SO WHAT IS THIS ABOUT?

There is clearly a need for CSS to be taken seriously by graphic artists. The Zen Garden aims to excite, inspire, and encourage participation. To begin, view some of the existing designs in the list. Clicking on any one will load the style sheet into this very page. The code remains the same, the only thing that has changed is the external .css file. Yes, really.

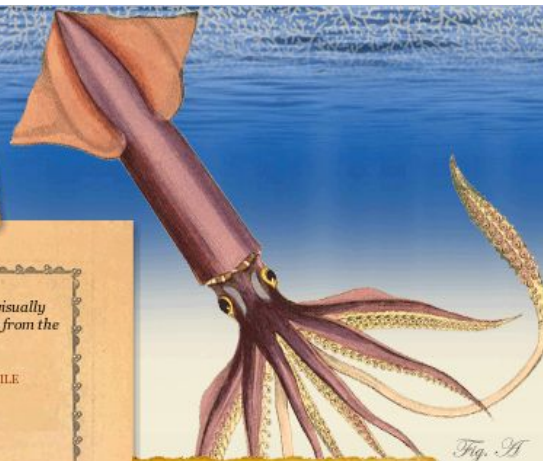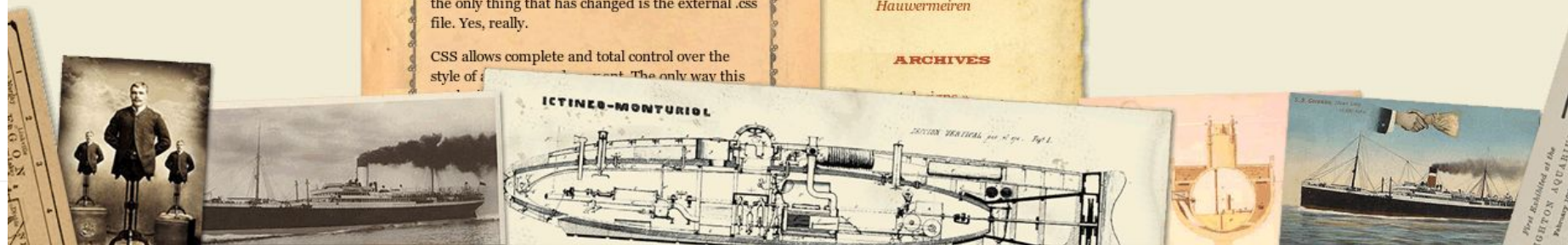CSS allows complete and total control over the style of a hypertext document. The only way this can be illustrated in a way that gets people excited is by demonstrating what it can truly be, once the reins are placed in the hands of those able to create beauty from structure. To date, most examples of neat tricks and hacks have been demonstrated by structurists and coders. Designers

CSS**OCEAN**GARDEN
THE BEAUTY OF CSS DESIGN

A demonstration of what can be accomplished visually through CSS based design. Select any style sheet from the list to load it into this page.

Download the sample html file and css file

THE **ROAD** TO **ENLIGHTENMENT**

Littering a dark and dreary road lay the past relics of browser-specific tags, incompatible DOMs, and broken CSS support.

Today, we must clear the mind of past practices. Web enlightenment has been achieved thanks to the tireless efforts of folk like the W3C, WaSP and the major browser creators.

The css Zen Garden invites you to relax and meditate on the important lessons of the masters. Begin to see with clarity. Learn to use the (yet to be) time-honored techniques in new and invigorating fashion. Become one with the web.

SO **WHAT** IS THIS **ABOUT**?

There is clearly a need for CSS to be taken seriously by graphic artists. The Zen Garden aims to excite, inspire, and encourage participation. To begin, view some of the existing designs in the list. Clicking on any one will load the style sheet into this very page. The code remains the same, the only thing that has changed is the external .css file. Yes, really.

CSS allows complete and total control over the style of a hypertext document. The only way this can be illustrated in a way that gets people excited is by demonstrating what it can truly be, once the reins are placed in the hands of those able to create beauty from structure. To date, most examples of neat tricks and hacks have been demonstrated by structurists and coders. Designers have yet to make their mark. This needs to change.

SELECT A DESIGN

UNDER THE SEA!
by Eric Stoltz

MAKE 'EM PROUD
by Michael McAghon And Scotty Reifsnyder

ORCHID BEAUTY
by Kevin Addison

OCEANSCAPE
by Justin Gray

CSS CO., LTD.
by Benjamin Klemm

SAKURA
by Tatsuya Uchida

KYOTO FOREST
by John Politowski

A WALK IN THE GARDEN
by Simon Van Hauwermeiren

ARICHIVES

next designs »
View All Designs

RESOURCES

View This Design's CSS
CSS Resources
FAQ
Submit a Design
Translations

CSS Zen Garden

A demonstration of what can be accomplished visually through CSS-based design. Select any stylesheet from the list to load it into this page.

Select a Design

- **Under the Sea!**
  by Eric Stoltz

- **Make 'em Proud**
  by Michael McAghon and Scotty Reifsnyder

- **Orchid Beauty**
  by Kevin Addison

- **Oceanscape**
  by Justin Gray

- **CSS Co., Ltd.**
  by Benjamin Klemm

- **Sakura**
  by Tatsuya Uchida

- **Kyoto Forest**
  by John Politowski

- **A Walk in the Garden**
  by Simon Van Hauwermeiren

Archives

- **next designs »**
- **View All Designs**

Resources

- **View This Design's CSS**
- **CSS Resources**

The Road to Enlightenment

Littering a dark and dreary road lay the past relics of browser-specific tags, incompatible DOMs, and broken CSS support.

Today, we must clear the mind of past practices. Web enlightenment has been achieved thanks to the tireless efforts of folk like the W3C, WaSP and the major browser creators.

The css Zen Garden invites you to relax and meditate on the important lessons of the masters. Begin to see with clarity. Learn to use the (yet to be) time-honored techniques in new and invigorating fashion. Become one with the web.

So What is This About

There is clearly a need for CSS to be taken seriously by graphic artists. The Zen Garden aims to excite, inspire, and encourage participation. To begin, view some of the existing designs in the list. Clicking on any one will load the style sheet into this very page. The code remains the same, the only thing that has changed is the external .css file. Yes, really.

CSS allows complete and total control over the style of a hypertext document. The only way this can be illustrated in a way that gets people excited is by demonstrating what it

Participation

Graphic artists only please. You are modifying this page, so strong CSS skills are necessary, but the example files are commented well enough that even CSS novices can use them as starting points. Please see the CSS Resource Guide for advanced tutorials and tips on working with CSS.

You may modify the style sheet in any way you wish, but not the HTML. This may seem daunting at first if you've never worked this way before, but follow the listed links to learn more, and use the sample files as a guide.

> " *A demonstration of what can be accomplished visually through CSS-based design. Select any style sheet from the list to load it into this page.* "

The Road to Enlightenment

Littering a dark and dreary road lay the past relics of browser-specific tags, incompatible DOMs, and broken CSS support.

Today, we must clear the mind of past practices. Web enlightenment has been achieved thanks to the tireless efforts of folk like the W3C, WaSP and the major browser creators.

The css Zen Garden invites you to relax and meditate on the important lessons of the masters. Begin to see with clarity. Learn to use the (yet to be) time-honored techniques in new and invigorating fashion. Become one with the web.

So What is This About?

There is clearly a need for CSS to be taken seriously by graphic artists. The Zen Garden aims to excite, inspire, and encourage participation. To begin, view some of the existing designs in the list. Clicking on any one will load the style sheet into this very page. The code remains the same, the only thing that has changed is the external .css file. Yes, really.

CSS allows complete and total control over the style of a hypertext document. The only way this can be illustrated in a way that gets people excited is by demonstrating what it can truly be, once the reins are placed in the hands of those able to create beauty from structure. To date, most examples of neat tricks and hacks have been demonstrated by structurists and coders. Designers have yet to make their mark. This needs to change.

Participation

Graphic artists only please. You are modifying this page, so strong CSS skills are necessary, but the example files are commented well enough that even CSS novices can use them as starting points. Please see the CSS Resource Guide for advanced tutorials and tips on working with CSS.

You may modify the style sheet in any way you wish, but not the HTML. This may seem daunting at first if you've never worked this way before, but follow the listed links to learn more, and use the sample files as a guide.

Download the sample html file and css file to work on a copy locally. Once you have completed your masterpiece (and please, don't submit half-finished work) upload your .css file to a web server under

CSS ZEN GARDEN

Select a design

Archives

CSS ZEN Garden

the beauty of css design

A demonstration of what can be accomplished visually through CSS-based design. Select any style sheet from the list to load it into this page.

Download the sample *html file* and *css file* ↓

The Road to Enlightment

Littering a dark and dreary road lay the past relics of browser-specific tags, incompatible DOMs, and broken CSS support.

Today, we must clear the mind of past practices. Web enlightenment has been achieved thanks to the tireless efforts of folk like the W3C, WaSP and the major browser creators.

The css Zen Garden invites you to relax and meditate on the important lessons of the masters. Begin to see with clarity. Learn to use the (yet to be) time-honored techniques in new and invigorating fashion. Become one with the web.

So What is This About?

There is clearly a need for CSS to be taken seriously by graphic artists. The Zen Garden aims to excite, inspire, and encourage participation. To begin, view some of the existing designs in the list. Clicking on any one will load the style sheet into this very page. The code remains the same, the only thing that has changed is the external .css file. Yes, really.

CSS allows complete and total control over the style of a hypertext document. The only way this can be illustrated in a way that gets people excited is by demonstrating what it can

→ Select a Design:

Under the Sea!
by Eric Stoltz

Make 'em Proud
by Michael McAghon and Scotty Reifsnyder

Orchid Beauty
by Kevin Addison

Oceanscape
by Justin Gray

CSS Co., Ltd.
by Benjamin Klemm

Sakura
by Tatsuya Uchida

Kyoto Forest
by John Politowski

A Walk in the Garden
by Simon Van Hauwermeiren

THE BEAUTY OF CSS-BASED DESIGN

A demonstration of what can be accomplished visually through CSS-based design.

CSS ZEN GARDEN

THE ROAD TO Enlightenment

Littering a dark and dreary road lay the past relics of browser-specific tags, incompatible DOMs, and broken CSS support.

Today, we must clear the mind of past practices. Web enlightenment has been achieved thanks to the tireless efforts of folk like the W3C, WaSP and the major browser creators.

The css Zen Garden

SELECT A DESIGN

FLORAL TOUCH
by *Jadas Jimmy*

ELEGANCE IN SIMPLICITY
by *Mani Sheriar*

DAZZLING BEAUTY
by *Deny Sri Supriyono*

DARK ROSE
by *Rose Fu*

LEGGO MY EGO
by *Jon Tan*

LUGOZEE
by *Viallon Pierre-Antoine*

THE DIARY
by *Alexander Shabuniewicz*

LONELY FLOWER
by *Mitja Ribic*

ARCHIVES

next designs »

« previous designs

SO WHAT IS THIS ABOUT?

Download the sample html file and css file

There is clearly a need for CSS to be taken seriously by graphic artists. The Zen Garden aims to excite, inspire, and encourage participation. To begin, view some of the existing designs in the list. Clicking on any one will load the style sheet into this very page. The code remains the same, the only thing that has changed is the external .css file. Yes, really.

CSS allows complete and total control over the style of a hypertext document. The only way this can be illustrated in a way that gets people excited is by demonstrating what it can truly be, once the reins are placed in the hands of those able to create beauty from structure. To date, most examples of neat tricks and hacks have been demonstrated by structurists and coders. Designers have yet to make their mark. This needs to change.

PARTICIPATION

Graphic artists only please. You are modifying this page, so strong CSS skills are necessary, but the example files are commented well enough that even CSS novices can use them as starting points. Please see the CSS Resource Guide for advanced tutorials and tips on

CSS in HTML documents

Linked

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

Header style

```
...
<head>
      <style type="text/css">
          a {
             color: #F00;
          }

          div.ingress {
             font-weight: bold;
          }
      </style>
   </head>
...
```

CSS in HTML elements

Inline

```
<body>

    <div style="font-weight: bold;">
        This text will be bold
    </div>

</body>
```

This may not be a good idea.

If you decide to do this, think about how it would have affected Zen Garden if they did so.
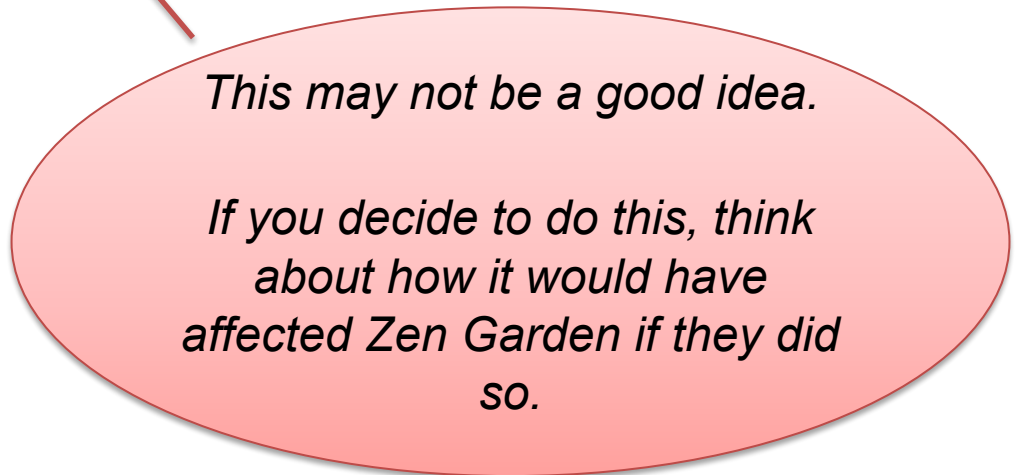
CSS3

- A new standard for CSS

- Includes specifications for rounded edges, gradients, shadows, rotation, Animation and etc.

- Only new web browsers support CSS3.

Antoine Sabot-Durand @antoine_sd · 23h

In Java, everything is an object.

In Clojure, everything is a list.

In Javascript, everything is a terrible mistake.
(Via @mabareghi)

JAVASCRIPT

The mighty Javascript

JavaScript ("JS" for short) is a full-fledged dynamic programming language that, when applied to an HTML document, can provide dynamic interactivity on websites. It was invented by Brendan Eich, co-founder of the Mozilla project, the Mozilla Foundation, and the Mozilla Corporation.

https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web/JavaScript_basics

Some facts

- JavaScript (JS) is a lightweight, interpreted, programming language with first-class functions.
- Many non-browser environments also use it, such as node.js and Apache CouchDB.
- JS is a prototype-based, multi-paradigm, dynamic scripting language, supporting object-oriented, imperative, and declarative (e.g. functional programming) styles.
- JS is not Java but syntactically close to C-family languages.

https://developer.mozilla.org/bm/docs/Web/JavaScript

Assignments and scope

- Variables can be defined globally everywhere
- Functions only have one scope
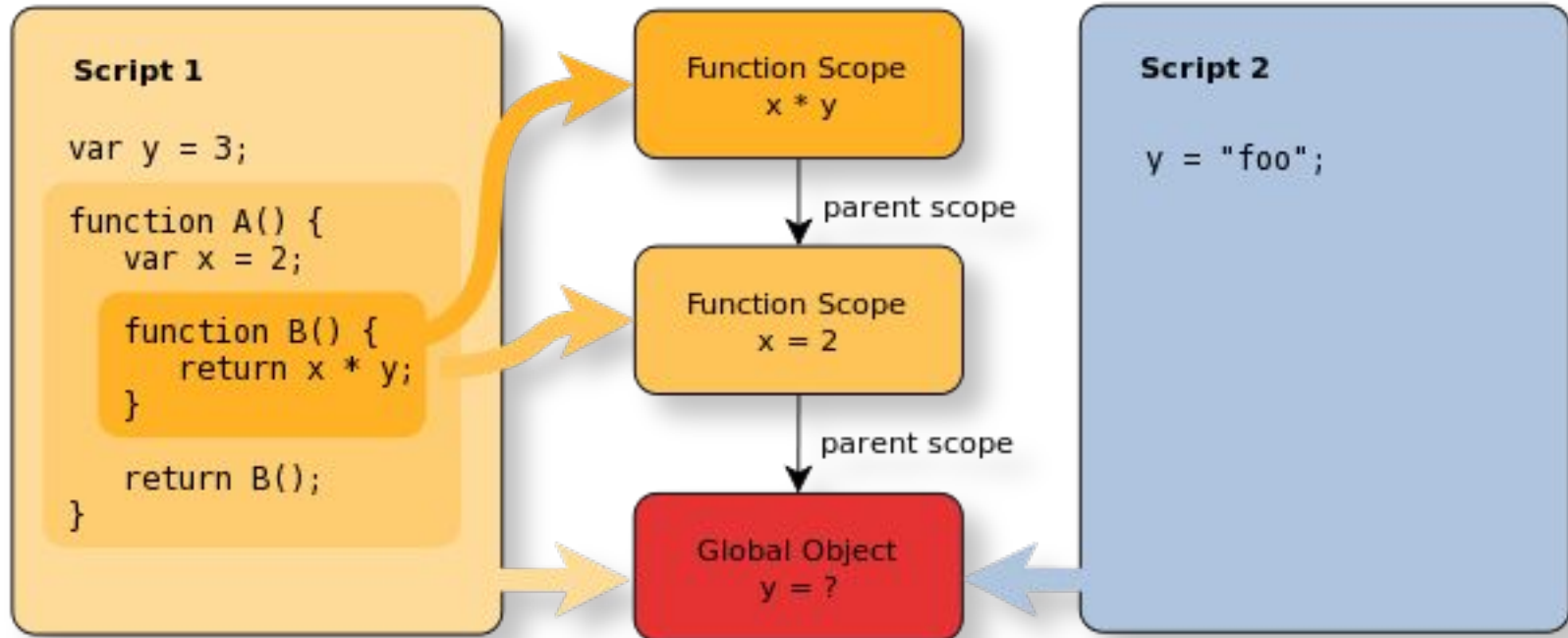
https://ringojs.org/documentation/module_and_global_scope/

Assignments and scope

```
function functionA() {
    variable = "Hello, world";    // Declared globally
}

function functionB() {
    return variable;
}

functionA();
functionB();     // Returns "Hello, world"
```

Assignments and scope

```
function functionA() {
    var variable = "Hello, world"; //Declared locally
}

function functionB() {
    return variable;
}

functionA();
functionB(); // ReferenceError: variable is not defined
```

Assignments and scope

```
var persons = new Array();
persons["Jakob"] = "Pogulis";
persons["Marcus"] = "Bendtsen";

function functionC() {
    for (p in persons) {
        alert(p);
    }
}

functionC();
alert(p);
```

Assignments and scope

```
var persons = new Array();
persons["Jakob"] = "Pogulis";
persons["Marcus"] = "Bendtsen";

function functionC() {
    for (var p in persons) {
        alert(p);
    }
}

functionC();
alert(c);
```

Assignments and scope

```
var persons = new Array();
persons["Jakob"] = "Pogulis";
persons["Marcus"] = "Bendtsen";

function functionC() {
    for (var p in persons) {
        alert(p);
    }

    alert(c);
}

functionC();
```

Assignments and scope

```
function functionD() {
    if (1 == 1) {
        var variable = "Hello, world";
    }

    alert(variable);
}

functionD();


// No block scope in Javascript. There is only function
scope.
```

Object-oriented programming

- JavaScript uses prototyping, unlike Java/Python/C++.

- JavaScript supports inheritance from one (1) prototype only.

- Regular object oriented programming in JavaScript is possible, but requires more memory since every object defines its own functions.

Prototyping

- Functions correspond to classes.

- Functions can contain functions.

- Classes can be extended after objects have been created.

- Objects can be extended after they have been created.

Prototyping

```
function Animal() {          // Constructor
    alert('Animal::Constructor');
}

Animal.prototype.talk = function() {  // Method
    return undefined;
}

var myAnimal = new Animal();
alert(myAnimal.talk());

Animal.prototype.scream = function() {
    return undefined;
}

alert(myAnimal.scream());
```

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Operators/new

Prototyping

```
function Animal() {          // Constructor
    alert('Animal::Constructor');
}

Animal.prototype.talk = function() {   // Method
    return "sound";
}

var myAnimal = new Animal();

Animal.prototype.whisper = function() {
    return "....";
}

myAnimal.scream = function() {
    return "SOUND!";
}

alert(myAnimal.whisper());
alert(myAnimal.talk());
alert(myAnimal.scream());
```

Callbacks

- Functions as arguments to functions.

- Delegates the responsibility to catch data and events to the called function.

- Large part of JavaScript and third-party libraries.

"I give you my phone number so you can call me when you have downloaded the next issue of The Newsroom (2012) [from iTunes]…"
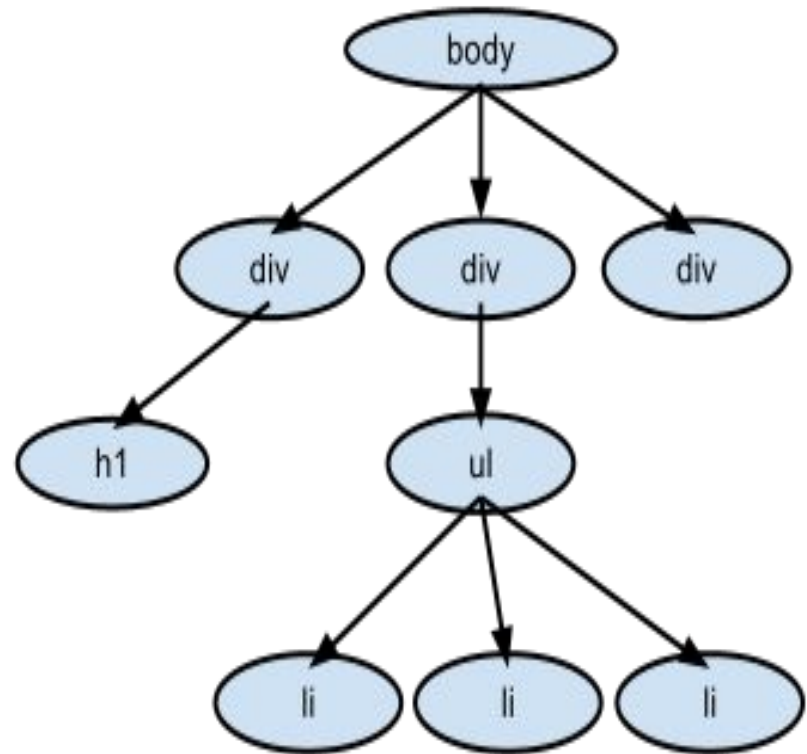
Document Object Model(DOM)

What is the HTML DOM?

The HTML DOM is a standard **object** model and **programming interface** for HTML. It defines:

- The HTML elements as **objects**
- The **properties** of all HTML elements
- The **methods** to access all HTML elements
- The **events** for all HTML elements

In other words: **The HTML DOM is a standard for how to get, change, add, or delete HTML elements.**

Document Object Model(DOM)

With the object model, JavaScript gets all the power it needs to create dynamic HTML:
- JavaScript can change all the HTML elements in the page
- JavaScript can change all the HTML attributes in the page
- JavaScript can change all the CSS styles in the page
- JavaScript can remove existing HTML elements and attributes
- JavaScript can add new HTML elements and attributes
- JavaScript can react to all existing HTML events in the page
- JavaScript can create new HTML events in the page

Navigating the DOM

- `getElementById('param')` returns one (1) element if there exists an element with the provided ID.

- `getElementsByTagName('param')` returns a list of elements if there exists one or more elements with a given name.

- `getElementsByClassName('param')` returns a list of elements if there exists one or more elements with a given class-name.

Node Operations

- `element.childNodes` returns a list of all nodes directly under an element in the DOM-tree.

- `element.parentNode` returns the node directly above an element in the DOM-tree.

- `element.nextSibling` returns the node directly to the right and at the same level as `element` in the DOM-tree.

- `element.previousSibling` returns the node directly to the left and at the same level as `element` in the DOM-tree.

Modifying the DOM-tree

- `document.createElement('param')` creates a new element.

- `document.createTextNode('param')` creates a new TextNode.

- `element.appendChild(element)` places the specified element last in the list of nodes directly under the specified element.

- `element.removeChild(element)` removes an element from the list of nodes directly under the specified element.

New Features

HTML5

THINGS AREN'T ALWAYS #000000 AND #FFFFFF

The most interesting new HTML5 elements

- New **semantic elements** like <header>, <footer>, <article>, and <section>.
- New **attributes of form elements** like number, date, time, calendar, and range.
- New **graphic elements**: <svg> and <canvas>.
- New **multimedia elements**: <audio> and <video>.

https://www.w3schools.com/html/html5_intro.asp
https://www.w3schools.com/html/html5_new_elements.asp

HTML5 new technologies

- Web Workers
- Audio and video capabilities
- History API
- Web Sockets
- Camera API
- Geolocation and detecting device orientation
- Canvas and SVG
- WebGL
- Web Storage
- ...

https://developer.mozilla.org/en-US/docs/Web/Guide/HTML/HTML5

HTML5 Demo

https://html5demos.com/

Web testing frameworks

SELENIUM

Selenium – A tool for testing web sites

Use a programming language (e.g., Python) to write code for describing different flows through the web site.

Automated testing of web sites can save lots of time, and therefore reduce costs.

Open web page

Find element with id 'q'

Type the word "Cheese" in the element with id 'q'

Press enter

Wait at most 10 seconds for a web page with a title that starts with the word cheese.

Selenium – A tool for testing web sites

```
import selenium
from selenium import webdriver
from selenium.common.exceptions import
TimeoutException
from selenium.webdriver.support.ui import WebDriverWait
import time

driver = webdriver.Firefox()

driver.get("http://www.google.com")
```

Selenium – A tool for testing web sites

.....

inputElement = driver.find_element_by_name("q")

inputElement.send_keys("Cheese!")

inputElement.submit()

....

Selenium – A tool for testing web sites

```
...

print driver.title

try:
   WebDriverWait(driver, 10).until(
      lambda driver : driver.title.lower().startswith("cheese!")
      )


   print driver.title

finally:
   driver.quit()
```

http://epicmediainc.com/8-reasons-responsive-web-design-will-increase-profit-business/

Responsive Web Design

RWD

- Responsive Web Design makes your web page look good on all devices (desktops, tablets, and phones).
- Responsive Web Design is about using HTML and CSS to resize, hide, shrink, enlarge, or move the content to make it look good on any screen.

Demo:

https://www.w3schools.com/html/tryit.asp?filename=tryhtml_responsive

RWD vs dedicated mobile version

- Dedicated: More expensive to maintain but faster to load.
- RWD: Cheaper to maintain but slower to load.

https://www.atilus.com/mobile-optimized-vs-responsive-websites/

Responsive Grid-view

- Many web pages are based on a grid-view, which means that the page is divided into columns.
- A responsive grid-view often has 12 columns, and has a total width of 100%, and will shrink and expand as you resize the browser window.

https://www.w3schools.com/css/css_rwd_grid.asp

Responsive Grid-view

One of the best implementations:

http://www.bbc.com/

About Lesson 01

- **Implementing the front-end for "The PhoneBook", a sample web application.**
- **Reviewing lab 01**

Thanks for listening!