

WEB MAPPING TECHNOLOGY – AN OVERVIEW

Interactive Map Design | Spring 2018

How does it work?

- Interactive design technologies like Javascript
- Map services like Google maps, OSM
- Spatial data types (KML, shp, GeoJSON)
- APIs (application program interface)
- Mash-ups (created by combining data or functionality from different sources)

Let's look at the essentials of....



- Web coding
- Spatial data types for the web
- Tools & services for creating web maps



The gist on coding

HTML, Javascript, & CSS



If you want to design Web Maps, you have to learn how to read & write some code.

You ***do not*** have to become a coding master.

With a little bit of coding skills, you can do some incredible things.

HTML

Hypertext Markup Language

- Backbone of the web – the language in which content is provided to web browsers
- Syntax works via tags
- Used almost solely for content organizing & delivery
- Styling & Interactivity is left to CSS & Javascript

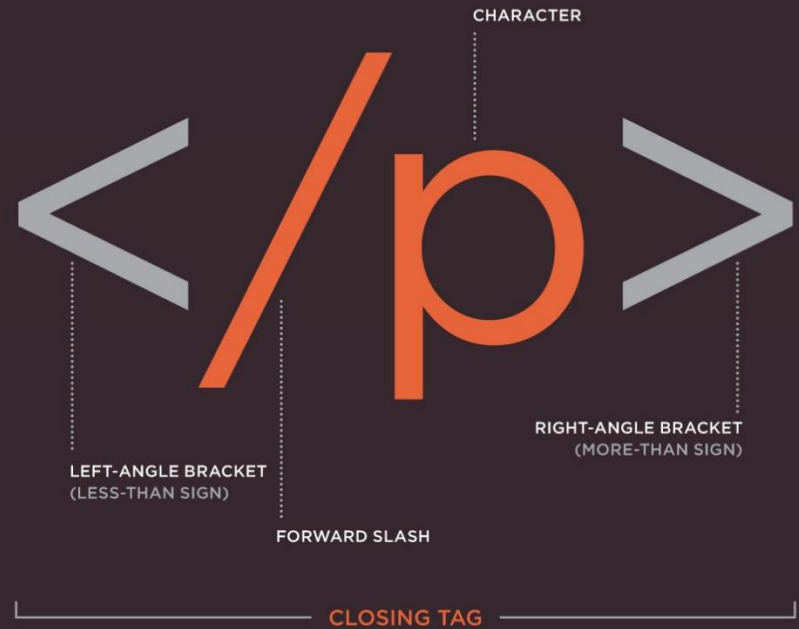
A CLOSER LOOK AT TAGS



The characters in the brackets indicate the tag's purpose.

For example, in the tags above the `p` stands for paragraph.

The closing tag has a forward slash after the `<` symbol.



The terms "tag" and "element" are often used interchangeably.

Strictly speaking, however, an element comprises the opening

tag and the closing tag and any content that lies between them.

What Beautiful HTML Code Looks Like

DOCTYPE Properly Declared

It looks like a lot of gibberish, but DOCTYPEs are important. They not only allow your code to validate, but they tell browsers things about how to render your page. Simple <html> tags don't cut it.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

```
<html xmlns="http://www.w3.org/1999/xhtml">
```

```
<head>
  <title>Urban and Regional Planning</title>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
  <link rel="stylesheet" type="text/css" href="css/print.css" media="print" />
  <link rel="stylesheet" type="text/css" href="css/style.css" media="screen, projection" />
  <script src="scripts/fun.js" type="text/javascript"></script>
</head>
```

Tidy Head Section

Title is set. Character set declared. Stylesheets linked (including a print stylesheet!). Scripts linked and NOT included in full. External files have their own related folders (e.g. "CSS" & "Scripts")

Body IDed

Putting an ID on your body allows you to create CSS properties that are unique to that page. For instance, you may want your <h2> tags to look different on the homepage. In your CSS you can write: #home h2 {} to accomplish this and not affect <h2> tags elsewhere.

```
<body id="home">
```

```
  <div id="page-wrap">
```

```
    <?php include_once("menu.html") ?>
```

```
    <div id="main-content">
```

```
      
```

```
      <div id="search-area">
        <input value="Search..." onfocus="this.value='';" />
      <div style="clear: both;"></div>
    </div>
```

```
    <?php include_once("left-sidebar.html") ?>
```

```
    <div id="right-content">
```

```
      <div id="right-sidebar">
        <div class="news-box">
          <?php include_once("news.html") ?>
        </div>
```

```
        <div class="events-box">
          <?php include_once("events.html") ?>
        </div>
      </div>
```

```
      <h1>Welcome to the URPL!</h1>
```

```
      <h2>Who we are:</h2>
```

```
      <p>Over the course of the past 50 or so years, we in the <strong>Department of Urban and Regional Planning</strong> have been active in the core missions of teaching, research, and outreach. Since inception in the 1960s, we have granted about 1000 graduate degrees and forwarded a progressive and interdisciplinary approach to the theory and practice of planning. As a department of both the College of Letters and Science and the College of Agricultural and Life Sciences, our faculty, affiliates, and students provide a bridge between the array of academic disciplines and knowledge bases necessary to address key planning problems found in the real-world.</p>
```

```
      <h2>Our Goals:</h2>
```

```
      <p>The department has three primary goals. First, we actively prepare qualified graduate students to become competent, creative and effective practicing planners. Second, we contribute to knowledge in the field of planning through scholarly and applied research. And third, we undertake professional planning activities and provide service in collaboration with the University of Wisconsin - Extension, a variety of public agencies, planning consulting firms and other private and non-profit sector organizations.</p>
```

```
    </div>
  </div>
```

```
  <?php include_once("footer.html") ?>
```

```
</div>
```

```
</body>
</html>
```

Semantically Clean Menu

```
<div id="menu">
  <ul>
    <li><a href="index.php">Home</a></li>
    <li><a href="about.php">About</a></li>
    <li><a href="contact.php">Contact</a></li>
  </ul>
</div>
```

Main DIV for all Page Content

Putting all the content of your page into one main "wrap" DIV gives you lots of control right off the bat. There is where you can set the width of your page for a fixed width site or maximums and minimums for a fluid width site.

Common Content INCLUDED

A lot of web content is common from page to page. Think menu bars, sidebars, footers, "boxes", etc. This kind of content should be dynamically loaded. Either from a database or with simple PHP include statements.

Proper Ending Tags

You started strong, now end strong. Don't be lazy and exclude closing tags for any element, even if the page renders OK without them.

Content, Content, Content

This is where your content belongs, so go nuts. Remember to keep your paragraphs distinct and in <p> tags. Use lists where appropriate. Use codes like © for © symbols. Don't go overboard with
 tags, that's sloppy formatting.

Important Content First

It is best if your most important content, like news and events, can be listed first in the HTML. If your sidebar is just navigation or less important content, it is best if it comes last in the HTML.

Code is Tabbed into Sections

If each section of code is tabbed in once, the structure of the code is much more understandable. Code that is all left-justified is horrific to read and understand.

Hierarchy of Header Tags

Use header tags as they were designed, to create titles for sections and signify their position in the content hierarchy.

No Styling!

Your HTML should be focused on structure and content, not styling! Keep all of your styling in your CSS, there should be no deprecated tags in site.

HTML5

- An attempt to create a framework of open-source technologies that can be used together to mimic what plug-ins could really only do (i.e. Flash player) for a dynamic web
- Comprised of 3 separate languages (despite being called HTML5): HTML, CSS, Javascript
- Possible to design HTML with a text editor making it more accessible & open

CSS

Cascading **Style Sheets**

- ❑ Describes how HTML elements are to be displayed
- ❑ HTML provides & organizes content; CSS styles the content for the browser
- ❑ From a web design standpoint, it saves time by controlling the layout of multiple web pages all at once
- ❑ Defines styles for your web pages, including the design, layout & variations in display for different devices & screen sizes

CSS solved a big problem

HTML is not intended to contain tags for formatting a web page. It is used to describe the **CONTENT** of a web page

```
<H1> This is a heading </H1>
```

Want to learn more about CSS?

<http://www.w3schools.com/css/default.asp> or
www.codecademy.com

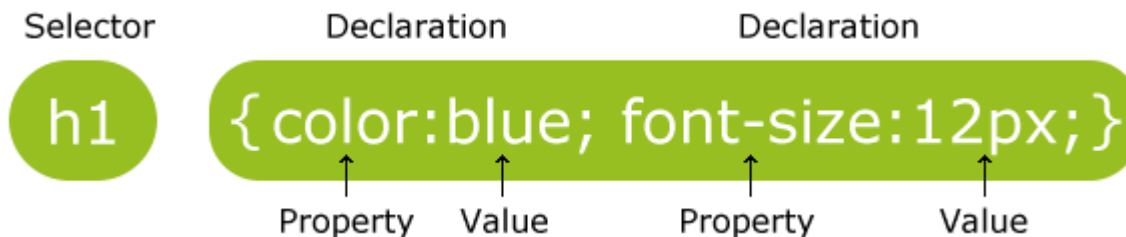
Why use CSS?

- This is what you will need to learn to make your web maps attractive
- Used to design the look & feel of the web map elements: borders, fills, outlines, drop shadows, shapes, layouts, etc.

How does CSS work?

“Selector” followed by a “Declaration”

- Selector. Selects which part of the HTML will be styled
- Declaration. Declares what properties of the selected HTML elements will be styled differently from the default HTML style and assigns them new values



CSS code example

```
1 /* GENERAL STYLES
2  *-----*/
3 html, body, form, fieldset, img, img a {
4     margin: 0;
5     padding: 0;
6     border: 0;
7 }
8 body {
9     color: #414141;
10    background: url(../images/bg.jpg) repeat-x #ebe8df;
11    font-family: Arial, Helvetica, sans-serif;
12    line-height: 120%;
13    font-size: 12px;
14 }
15
16 a:link, a:visited {
17     color: #685966;
18     text-decoration: underline;
19 }
20 a:hover {
21     color: #2b212c;
22 }
23 .article_separator {
24     line-height: 5px;
25     height: 5px;
26     font-size: 5px;
27 }
28 /* SITE WIDTH
29  *-----*/
30 .rht_container {
31     width: 1020px;
32     margin: 0 auto;
33     margin-top: 25px;
```

Javascript

- An object-oriented scripting language that is used to target specific, predefined objects (map elements), and manipulate them under certain defined conditions
- In layman's term - it makes maps interactive
- Not a plug-in script – it is read & run by the web browser and therefore on the user's computer.

Javascript

- Allows you to write functions or methods, that directly manipulate HTML content
 - ▣ Interact and update the web page content without reloading it
 - ▣ Animate parts of the web map
 - ▣ Validate form information
 - ▣ Fetch external data

Javascript

- A robust language that could take a while to learn, BUT once you write a script that does something you can reuse it / modify it for use on all your maps!

```
function initialize() {
  var mapOptions = {
    zoom: 3,
    center: new google.maps.LatLng(0, -180),
    mapTypeId: google.maps.MapTypeId.TERRAIN
  };

  var map = new google.maps.Map(document.getElementById('map'), mapOptions);

  var flightPlanCoordinates = [
    new google.maps.LatLng(37.772323, -122.214897),
    new google.maps.LatLng(21.291982, -157.821856),
    new google.maps.LatLng(-18.142599, 178.431),
    new google.maps.LatLng(-27.46758, 153.027892)
  ];
  var flightPath = new google.maps.Polyline({
    path: flightPlanCoordinates,
    editable: true,
    strokeColor: '#FF0000',
    strokeOpacity: 1.0,
    strokeWeight: 2,
    map: map
  });

  var deleteMenu = new DeleteMenu();
```

To review....



HTML to define the content of web pages

CSS to specify the layout of web pages

JavaScript to program the behavior of web pages

API

Application Programming Interface

- Heavily used in web mapping to allow different computer programs to speak & interact with one another
- For example, API's allow the browser to use Javascript to communicate with an online mapping service

API

- Just to name a few....

- Bing maps

- Nokia

- Google

- MapQuest

- Leaflet

- Open layers

- Foursquare

- OpenStreetMap

- MapBox

- Carto

- ArcGIS

- <http://www.programmableweb.com/news/top-10-mapping-apis-google-maps-microsoft-bing-maps-and-mapquest/analysis/2015/02/23>



Spatial Data for the Web

Projections

You will never escape me!

- Most APIs are set up in Web Mercator (standard Internet projection, for better or for worse)
- It is possible to change the projection via coding
- If you are creating a fine-scale map, Web Mercator is probably not a good option; it is possible to use SVG (scalable vector graphic) rather than an API

KML

Keyhole **M**arkup **L**anguage

- Developed by Google as an XML language to represent spatial data
- XML is a tag-based language similar to HTML
- Easy to create and can be written & edited using a simple text editor
- KMZ are zipped KML files

GeoJSON

- One of the most common data types in web mapping
- Advantages include
 - ▣ Smaller file sizes
 - ▣ More intuitive for reading & editing
 - ▣ Easy to control & manipulate using Javascript, making it easy to import into many APIs

SVG

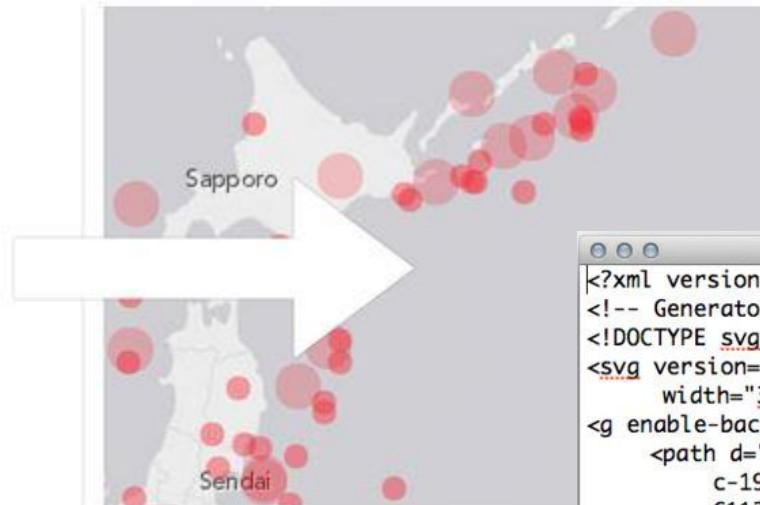
Scalable **V**ector **G**raphics

- An XML image format that does not pixelate with resizing
- Can be created using Adobe Illustrator and other online resources
- SVG files can be manipulated & animated using Javascript
- Great for creating non-Web Mercator thematic maps

SVG

Styling SVG with CSS

```
1 path[data-classification="0"] {
2   fill: none;
3   stroke: #f5cb00;
4   stroke-width: 3;
5   stroke-opacity: 1;
6   stroke-linecap: round;
7   stroke-linejoin: round;
8 }
9
10 path[data-classification="1"] {
11   fill: none;
12   stroke: #f67682;
13   stroke-width: 5;
14   stroke-opacity: 0.85;
15   stroke-linecap: round;
```



```
<?xml version="1.0" encoding="utf-8"?>
<!-- Generator: Adobe Illustrator 17.0.2, SVG Export
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN" "http
<svg version="1.1" id="Layer_1" xmlns="http://www.w3
width="300px" height="300px" viewBox="0 0 300
<g enable-background="new " >
  <path d="M220.784,150c0,19.533-6.914,36.214-20.
c-19.533,0-36.213-6.914-50.042-20.743C86.1
C113.787,86.129,130.468,79.215,150,79.215c
M209.463,150c0-16.421-5.806-30.436-17.419
l11.149,11.148l-7.865,8.644l-11.322-11.409
c11.61,11.609,25.626,17.415,42.047,17.415c
C204.68,177.514,209.463,164.636,209.463,15
c-5.043,5.013-11.106,7.52-18.193,7.52c-7.1
c-2.882-0.114-5.315-1.18-7.303-3.197c-1.98
c1.988-2.017,4.421-3.082,7.303-3.198v-38.1
V177.744z M124.072,158.729v-13.051c-4.207,
M152.333,111.281-17.977-4.148h-5.963v6.31
M170.311,128.824l-2.247-4.841-39.67,4.754
M171.435,177.744v-35.869h-3.458c0,6.167-0
c-6.108,0-9.162-5.532-9.162-16.595h-21.175
M138.678,151.728l-2.939,3.112l-2.506-3.11
</g>
</svg>
```

SVG resources

- https://commons.wikimedia.org/wiki/Category:Blank_SVG_maps_of_the_world
- <https://parall.ax/blog/view/2985/tutorial-creating-an-interactive-svg-map>
- <http://www.creativebloq.com/netmag/create-responsive-svg-image-maps-51411831>
- <https://pixelmap.amcharts.com/>
- <http://www.petercollingridge.co.uk/data-visualisation/introduction-svg-scripting-interactive-map>
- <http://design.tutsplus.com/tutorials/svg-files-from-illustrator-to-the-web--vector-20899>

Map Tiles

- Effective way to distribute detailed base maps that are generalized for numerous scales
- Break down large map down into rasterized squares or 'tiles'
- All tiles are the same size: 256 x 256 pixels
- Load much fast than one big map; only loads the tiles that are currently in view
- Each zoom level has its own set of tiles; number of tiles grows exponentially with each zoom level
- Rendered in advance then stored in a cache

Map Tiles – some drawbacks

- Hosting detailed tiles of large areas takes massive amounts of storage
- If something drastic changes with the base map, it can be time consuming to update

Map Tile Services

Do I have to make my own custom base layer tiles?

Of course not!

- There are many services out there that you can access: OpenStreetMap, Google, Carto, Mapbox
- Add additional style to your map with Mapbox or Carto to design your own tiles

Tiles

<http://tile.openstreetmap.org/17/35303/48503.png>

= name of the tile server

<http://tile.openstreetmap.org/17/35303/48503.png>

= zoom level

<http://tile.openstreetmap.org/17/35303/48503.png>

= x/y value, or the place in the grid where the tile lives



Tiles = Base map

Base map + data layers = web map!