

Web Design – Lecture-8

INSTRUCTOR: VIDHI KESARIA

Northeastern University
Seattle

Class Schedule

- ▶ 1) Website Architecture, Design, Strategy and Planning and Creating web pages – Semantic HTML4/5
- ▶ 2) Design using CSS (covers CSS3)
- ▶ 3) Responsive web design,
- ▶ 4) Making the web page Interactive - JavaScript (Datatypes, Inline, Embedded and external JavaScript, variables, operators, loops, functions, arrays, objects)
- ▶ 5) Working with DOM Model and JavaScript Events
- ▶ 6) Using JQuery & Hosting on AWS
- ▶ 7) Assignment presentation with demo.
- ▶ 8) Local storage, cookies, Personalization & wireframing for project
- ▶ 9) Design Style guide, Using LESS Preprocessor + Project work
- ▶ 10) Review styled website with actual content, learn about interactive javascript + + Project work
- ▶ 11) Performance + + Project work
- ▶ 12) SEO Techniques + + Project work
- ▶ 13) Accessibility Techniques & Tools + Project work
- ▶ 14) Written exams, Final project submission & Demo

Storing Information – Client side

What are cookies?

- ▶ A cookie is a text file stored on a user's computer.

Cookies allow you to personalize a Web site experience.

For example, you can greet return visitors by name or store their shipping address to automatically fill out online order forms when they place repeat orders.

What are the six parts of a cookie string?

- ▶ The six parts of a cookie string are:

name
value
expires
path
domain
secure

What are the parts of a cookie string?

- ▶ name and value

A name is used to reference the cookie so that you can retrieve the information stored in it.

The value is the actual information part of the cookie.

```
document.cookie="UserName=Paul;";
```

What are the parts of a cookie string?

▶ expires

If you want a cookie to exist for longer than a single user session—a single visit by the user to your Web site—you need to use `expires` to set an expiration date for the cookie.

```
document.cookie = "UserName=Paul;expires=Tue, 28 Dec 2020  
00:00:00 GMT; ";
```

What are the parts of a cookie string?

▶ path

Cookies are specific not only to a specific domain, but also to a particular path on that domain.

```
document.cookie = "UserName=Paul;expires=Tue, 28 Dec 2020  
00:00:00;path=/";
```


What are the parts of a cookie string?

- ▶ `secure`

The secure part of the cookie string is a Boolean value.

If it's set to true, the cookie will only be sent to a Web server that retrieves it using a secure channel. If it's set to false, the cookies will always be sent, regardless of the security.

Using Cookies – Javascript syntax

- ▶ Create cookie

- ▶ `setCookie("Name", "Bob", "", "");`

- ▶ Change a cookie

- ▶ `setCookie("Name", "Bobby", "", "");`

- ▶ Delete a cookie

- ▶ `setCookie("Name", "", "", "Mon, 1 Jan 1990 00:00:00");`

How do you read a cookie's value?

- ▶ You use `document.cookie` to retrieve a string containing information about the cookies that have been set.
- ▶ Only the name and value parts of a cookie can be retrieved.
- ▶ The `cookie` property allows you to retrieve only *all* the cookies set for a particular path and domain.

There's no simple way of just getting the value of a single cookie. To do this you'll have to use string manipulation techniques.

What are the limitations of cookies?

- ▶ The user may have disabled support for cookies in their browser.
- ▶ There is a limit to the number of cookies and the size of cookies:
 - ▶ for each domain, you can store up to 20 cookies.
 - ▶ Modern browsers have 300 cookie limit but vary
 - ▶ each name/value pair in a cookie can contain up to 4096 characters

Viewing cookies for validation

- ▶ Native Browser – <http://www.wikihow.com/View-Cookies>
 - ▶ Lets check out in chrome –
 - ▶ https://www.w3schools.com/js/js_cookies.asp
 - ▶ https://www.w3schools.com/js/tryit.asp?filename=tryjs_cookie_username
-
- ▶ You can also use web developer extension (its very handy for lot of other things as well) – its available for most of the browsers.

HTML Local Storage

- ▶ Before HTML5, application data had to be stored in cookies.
- ▶ With local storage, web applications can store data locally within the user's browser.
- ▶ Local storage is more secure, and large amounts of data can be stored locally, without affecting website performance.

Benefits of Local Storage

- ▶ Reduced network traffic
- ▶ Rapid display times
- ▶ Cache data from Remote Procedure Call (RPC)
- ▶ Load cached data on startup (faster startup)
- ▶ Save temporary state
- ▶ Restore state upon app re-entry
- ▶ Prevent work loss from network disconnects

HTML 5 Local Storage

- ▶ HTML local storage provides two objects for storing data on the client:
- ▶ `window.localStorage` - stores data with no expiration date
- ▶ `window.sessionStorage` - stores data for one session (data is lost when the browser tab is closed)
- ▶ Before using local storage, check browser support for `localStorage` and `sessionStorage`:
- ▶

```
if (typeof(Storage) !== "undefined") {  
    // Code for localStorage/sessionStorage.  
} else {  
    // Sorry! No Web Storage support..  
}
```


The localStorage Object

- ▶ The localStorage object stores the data with no expiration date. The data will not be deleted when the browser is closed, and will be available the next day, week, or year.
- ▶

```
// Store  
localStorage.setItem("lastname", "Smith");
```
- ▶

```
// Retrieve  
document.getElementById("result").innerHTML =  
localStorage.getItem("lastname");
```

The sessionStorage Object

- ▶ The sessionStorage object is equal to the localStorage object, **except** that it stores the data for only one session. The data is deleted when the user closes the specific browser tab
- ▶

```
if (sessionStorage.clickcount) {  
    sessionStorage.clickcount = Number(sessionStorage.clickcount) + 1;  
} else {  
    sessionStorage.clickcount = 1;  
}  
document.getElementById("result").innerHTML = "You have clicked the  
button " +  
sessionStorage.clickcount + " time(s) in this session.";
```

Using Local storage – Javascript syntax

- ▶ Three methods are important for the local storage `setItem()`, `getItem()` and `removeItem()`.
- ▶ Using single line code to store your data into your browser

To store data - `setItem()`

- `localStorage.setItem('FirstName', fName);`
- Name is key and the value is firstName
- **Example:**
- `function SetItem() { // save Value to local storage var fName = document.getElementById('Name').value; localStorage.setItem('FirstName', fName); }`

To get the stored item - `getItem()`

- `localStorage.getItem('FirstName');`
- **Example:**
- `function GetItem() { // Get Value from local storage var getStoredValue = localStorage.getItem('FirstName'); if (getStoredValue) { document.getElementById('Name').value = getStoredValue; } }`

To remove data from browser storage - `removeItem()`

- `localStorage.removeItem('FirstName');`
- **Example:**
- `function RemoveItem() { // Remove Value from local storage localStorage.removeItem('FirstName'); }`

Example

```
<!DOCTYPE html>
<html lang="en">
<head>
<title>HTML5 local storage</title>
<script type="text/JavaScript">
    function SetItem() { // save Value to local storage
        var fName = document.getElementById('Name').value;
        localStorage.setItem('FirstName', fName);
    }
    function GetItem() { // Get Value from local storage
        var getStoredValue = localStorage.getItem('FirstName');
        if (getStoredValue) {
            document.getElementById('Name').value = getStoredValue;
        }
    }
    function RemoveItem() { // Remove Value from local storage
        localStorage.removeItem('FirstName');
    }
</script>
<body>
Name : <input type="text" id="Name" />
<input type="button" value="save" onclick="SetItem()" />
<input type="button" value="Get" onclick="GetItem()" />
<input type="button" value="Remove" onclick="RemoveItem()" />
</body>
</html>
```

Cookies & Local Storage Summary

Cookies

▶ Pros

- ▶ Legacy support (it's been around forever)
- ▶ Persistent data
- ▶ Expiration dates

▶ Cons

- ▶ Each domain stores all its cookies in a single string, which can make parsing data difficult
- ▶ Data is unencrypted, which becomes an issue because... .. though small in size, cookies are sent with every HTTP request Limited size (4KB)
- ▶ SQL injection can be performed from a cookie

Local Storage

▶ Pros

- ▶ Support by most modern browsers
- ▶ Persistent data that is stored directly in the browser
- ▶ Same-origin rules apply to local storage data
- ▶ Is not sent with every HTTP request
- ▶ ~5MB storage per domain (that's 5120KB)

▶ Cons

- ▶ Not supported by anything before: IE 8 Firefox 3.5 Safari 4 Chrome 4 Opera 10.5 iOS 2.0 Android 2.0
- ▶ If the server needs stored client information you purposefully have to send it.

Why Personalization?

- ▶ **(74%) of online consumers get frustrated with websites when content appears that has nothing to do with their interests**
- ▶ **CTAs targeted to the user had a 42% higher view-to-submission rate** than calls-to-action that were the same for all visitors
- ▶ **Average a 19% uplift in sales**
- ▶ offers users content and experiences specific to their profile, interests, habits and behaviors
- ▶ **establishes customer loyalty**, which leads to **repeat visitors**, which leads to repeat sales, and ultimately, to higher profits.
- ▶ 5 times more likely to attract repeat visitors than non-customized sites.
- ▶ **Eg.** Amazon, which many consider the best of online retailers offering personalization, originates 35 percent of its product sales through personalized recommendations.

How does Personalization work?

- ▶ A personalized digital experience works by connecting technologies and algorithms to data stored in various repositories and comparing it to the user's profile to create a dynamic experience unique to that user.
- ▶ Two types of data:

Explicit (User must perform an action)
<ul style="list-style-type: none">• Filling out a form• Making a purchase• Favoriting an item

Implicit (Tracking technology gathers behavioral data)
<ul style="list-style-type: none">• Browsing activity• Searches• Location

Benefits of Personalization

- ▶ more relevant,
- ▶ more topical
- ▶ more inspirational

- ▶ That results in better
 - ▶ customer satisfaction,
 - ▶ better retention,
 - ▶ higher engagement
 - ▶ product sales

Lab work – Portfolio website personalization

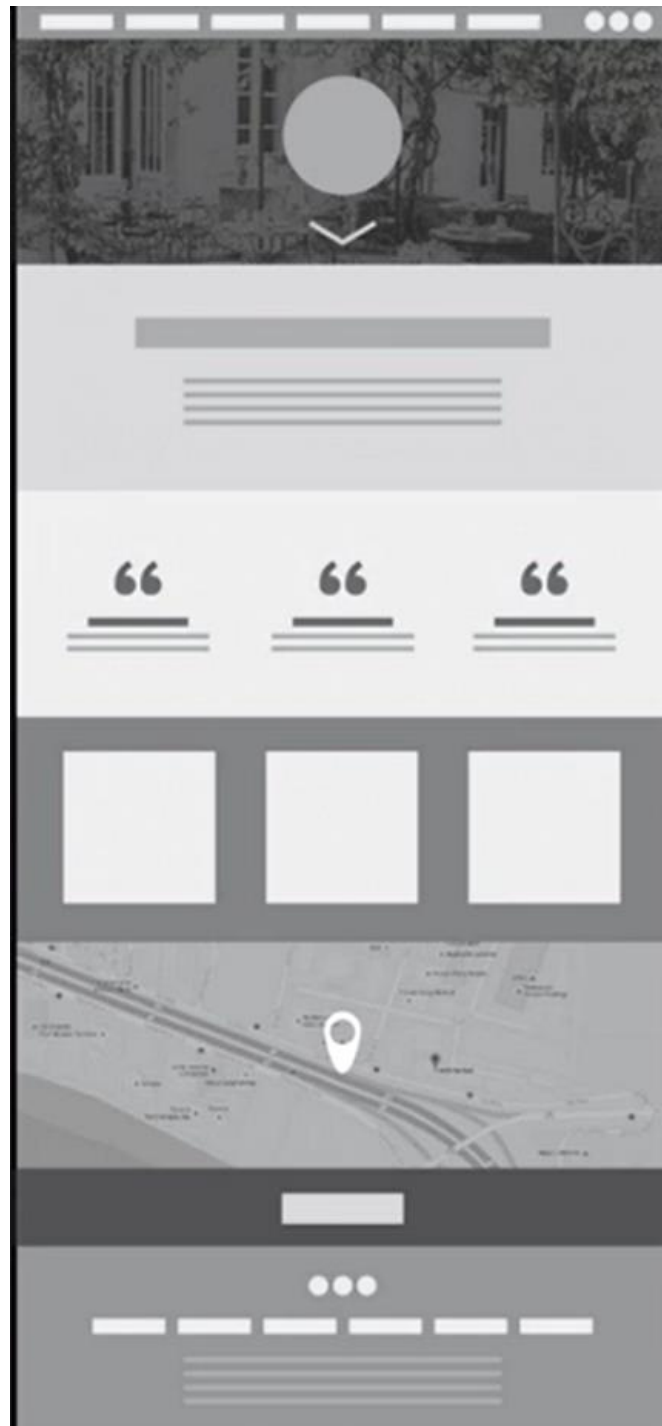
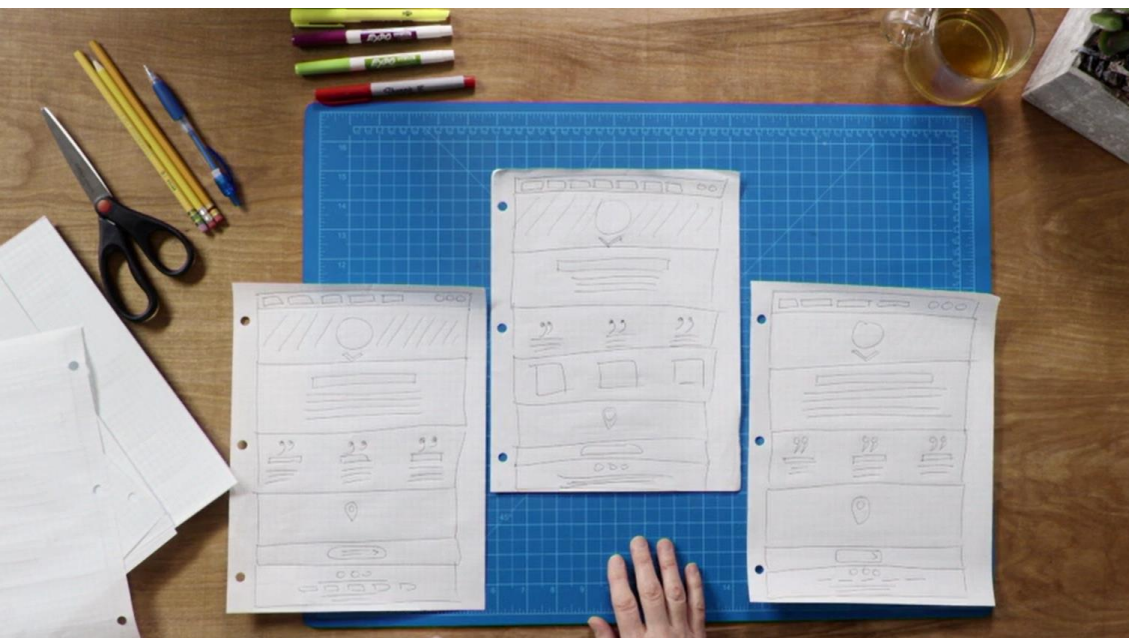
- ▶ Content personalization:
 - ▶ You have list of projects in your project page: Backend development (A, b, c, d), Front end development (E,f,g,x,y), Business related (H, J)
 - ▶ When a user clicks on view project details for project f – store the information for the project that was selected and the category it belonged to ("current project:"f", "category:"front-end-dev". You can use data- attributes to store this information in html tags
 - ▶ Next time when the user visits your portfolio, retrieve your information from the local storage, and based on the project that was previously seen by your employer, highlight those projects more, so in this case "Front end development" related projects. You can also order them to first in your project details page
 - ▶ Alternatively, you can also start with asking the user, what know of role they are looking for and provide them options – eg. Front end developer, ux designer, back end developer, software engineer, product manager) – Based on their selection, update your projects & skills that match to that role higher in order (you can use data-attribute to store that information). Next time they visit the site, you can chose default option.
- ▶ Design personalization
 - ▶ When a user choses a theme, persist that info in local storage and next time when they visit, chose that same theme
- ▶ Aggregate data personalization
 - ▶ Store the keyword every time there is a search, add weightage to that search
 - ▶ Depending on the weightage, update the order of your content on skills page – so the most searched is on top of the page (so not always ordered by date, but could be by skills)

Project work

- ▶ 5 mins break ---- will start at 12:20

Web Design Workflow – Structural Design

- ▶ Define the purpose
 - ▶ What is the goal of your website
 - ▶ Who your customers are their needs (Persona)
- ▶ Create sitemap outlines informational architecture
 - ▶ Think of user workflows
 - ▶ Create Simple diagram of how pages are organized and linked together -> Site Navigation & Flow
- ▶ Content Modelling & Features Based on workflow
- ▶ Wireframes
 - ▶ Visually demonstrate the structural framework for the hierarchy of information by order, size, functionality, and relative importance
 - ▶ Eg. might include a header with a logo, navigation, search bar, an animated slider, a signup form, top news stories, featured items, a call to action, the footer, and other interactive elements.
- ▶ Designing your site



- ✓ clear understanding of the goals
- ✓ hierarchy of information
- ✓ Outcomes
- ✓ and content

Web Design Workflow – Visual Design

- ▶ Visual guide that captures the mood or feel of a project.
 - ▶ your company name and branding info,
 - ▶ Assets
 - ▶ fonts and sample text,
 - ▶ color palette textures and patterns,
 - ▶ photos and illustrations.

The Love to Ride visual identity relies heavily on the use of colour. The palette consists of a complementary range of colours that work well in almost any combination.

Our brand colours

We have defined a strict colour palette which is restricted to the colours below. This gives the brand a very unique look and feel that enables users to be distinguished instantly.

These colours are never to be used as tints or shades. Only use the colours shown below in the exact shade. (Pantone, CMYK or RGB depending on your medium).

Tints are allowed only in the gradient makeup of the icon (see Page 6).



C0 M0 Y0 K0
RGB 255 255 255
Pantone Cool Grey 10



C47 M07 Y07 K18
RGB 205 205 205
Pantone Cool Grey 80



C56 M11 Y42 K05
RGB 100 100 100
Pantone Black 60



C58 M58 Y53 K58
RGB 54 54 54
Pantone Process Black 0



C0 M76 Y61 K0
RGB 220 68 87
Pantone 7100



C71 M3 Y27 K0
RGB 51 205 178
Pantone 31150



C5 M26 Y75 K0
RGB 234 218 96
Pantone 1280



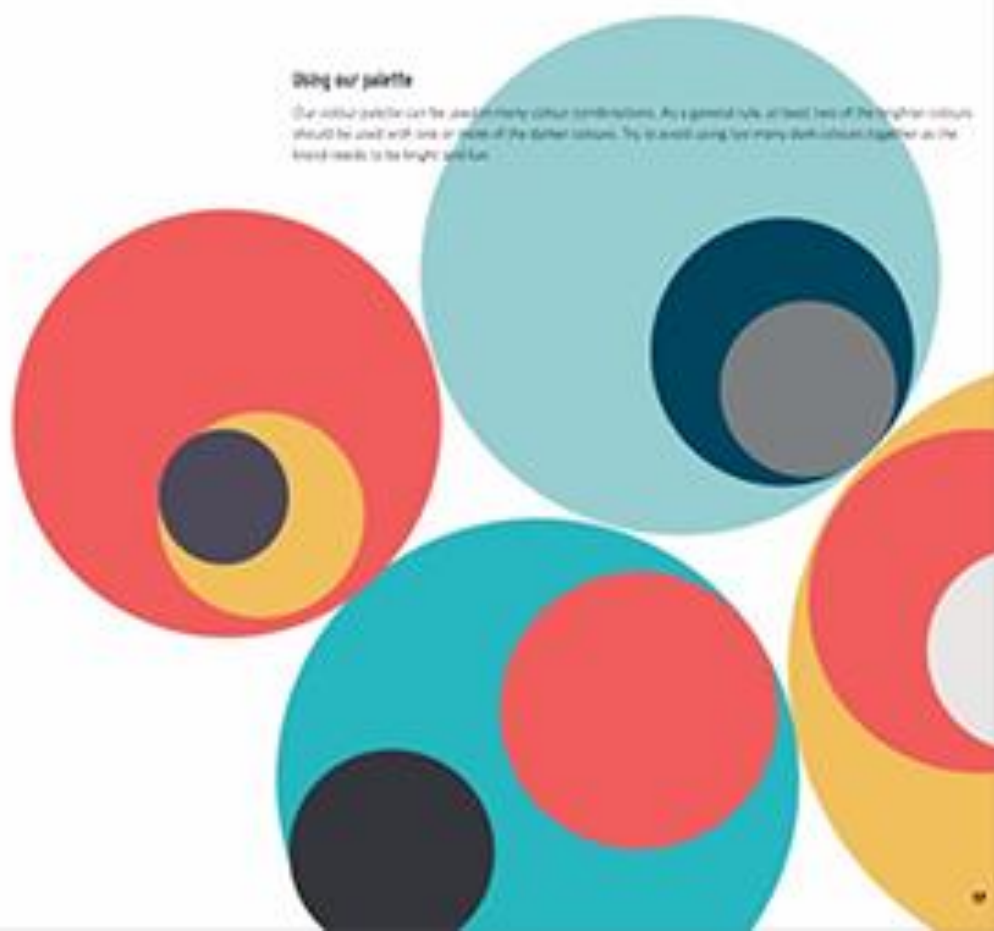
C36 M65 Y40 K34
RGB 51 51 218
Pantone 5480



C40 M3 Y18 K0
RGB 127 218 218
Pantone 28750

Using our palette

Our colour palette can be used in many colour combinations. As a general rule, at least two of the lighter colours should be used with one or more of the darker colours. Try to avoid using too many dark colours together as the brand needs to be bright and fun.



At this time

- ▶ Next steps:
 - ▶ Inserting content
 - ▶ Fine-tuning the layout.