

Manage and Design the Website

Planning

- Identify the **audience**
- State the **goals** of the web site
- Define **interaction** points
- Describe the site both **textually and visually**
- Identify and assess **similar sites**
- Provide a guide for **testing**

Storyboard

- Manage the content and position
- Included color
- Which part is permanent or not

Hi.



I WON'T WAIT LONG.
AT \$55,000, I'M PRICED TO GO.



Personal Site dot-com

[Home](#) [About](#) [Archives](#) [Writing](#) [Speaking](#) [Contact](#)

Li European lingues

Epsum factorial non deposit quid pro quo hic escorol. Olypian quarrels et gorilla congolium sic ad nauseum. Souvlaki ignitus carborundum e pluribus unum. Defacto lingo est igpay atinlay. Marquee selectus non provisio incongruous feline nolo contendre.

Hello, World!

File and naming

- You need a plan for **putting site** assets in proper place, so you and others who work on the site **can easily find and update the right file at a later date.**
- You need to **name web page files** so they convey **meaningful information to your visitors** as well as you and others who work on the site
 - Have a valid file extension, such as .html, .gif, or .pdf
 - Convey something about the source, contents, or nature of the file
 - Follow a logical and consistent scheme across similar files

- You need to give **meaningful and reliable titles** to your web pages so they are easily distinguished from one another in search results and browser history lists
 - Maintain a consistent format.
 - Use specific language.
 - Make page titles unique.
 - Keep page titles brief.

- You need to follow a dependable pattern for **naming variables in your web site scripts**.
 - Uses unique, concise terms
 - Limits abbreviations
 - Avoids reserved words in its programming language
 - Serves as a form of self-documentation

Layout using Table

- **HTML Table and CSS** is pretty important to built page layout



- And take a look to any webpage...you will see it how html table and css built the layout
- **The easiest way** and still working
- **Fixed-width layouts use absolute pixel** measurements to determine the width of the page
- **Not fixed width is liquid or expando**, which automatically resizes itself to fit the width of the visitor's browser window


```
<table width="760" border="0" cellpadding="0"
cellspacing="0">
  <tr>
    <td width="760" valign="top">Logo</td>
  </tr>
  <tr>
    <td width="760" valign="top">
```

```
<div id="nav" style="position: absolute; width: 200px;">
```

- Absolute

Flexible and Fixed Layout

- You need to determine which design format will do the best job of presenting your web site's content for its audience.
 - **A fixed-width** layout that locks page elements in place regardless of the browser window size
 - **A flexible**, or liquid, layout in which content blocks can be resized when the browser window size changes
 - **hybrid layout** that combines both fixed and flexible components

```
#column1 {  
    float: left;  
    width: 150px;  
    margin-left: 5px;  
    background-color: #CCCCCC; }
```

```
#column2 {  
    float: left;  
    width: 50%;  
    margin-left: 2%;  
    background-color: #FFFFCC;}
```

```
<div id="col1">...column 1 content ...</div>
```

```
<div id="col2">...column 2 content ...</div>
```

- You should design your web site only once, not each time you create a new page

Asymmetric Layout

- First, mark up the content with div elements using the id attributes that contain appropriate values representing their placement on the page

```
<div id="header">
  [...]
</div>
<div
id="columnSmall">
  [...]
</div>
<div id="columnMain">
  [...]
</div>
<div
id="columnMedium">
  [...]
</div>
```



- Next, use the position property in each column, setting the value to absolute while setting the placement of the columns with the left and top properties using percentages

```
#header {  
  position: absolute;  
  left: 65%;  
  top: 50%;  
  width: 125px;  
  font-size: small;  
}  
#columnSmall {  
  position: absolute;  
  left: 35%;  
  width: 15%;  
  top: 1%;  
  background: #fff;  
  font-size: small;  
}
```

```
#columnMain {  
  position: absolute;  
  left: 5%;  
  width: 45%;  
  top: 40%;  
  background: #fff;  
  text-align: justify;  
  border-width: 0;  
  font-size: large;  
}  
#columnMedium {  
  position: absolute;  
  left: 80%;  
  width: 20%;  
  top: 10%;  
  background: #fff;  
}
```

Color Schema

- Sesuaikan dengan tema, brand
- Rencanakan gradasi warna, buat library warna
- Konsisten dengan warna
 - Konsep mixing warna
 - Konsep filosofi warna

Tipografi

- Roman family → klasik, art, keterbacaan kurang bagus
- Serif family → tegas, keterbacaan bagus
- Mixing tipografi

Multimedia Content

- Image, Animation, Video
- Need more space, sometimes problem to be downloaded
- Make sure size is not reason of trouble
- Be aware of copyright.
- Better make by yourself
- Suggestion format: .gif, .png, .mp4, .flv, .swf

- Kuasai at least one editing tool for each media.
Ex: Photoshop (raster), corel (vector), audacity (audio), etc...
- Pilih multimedia content sesuai dengan isi webpage/ tema/ tujuan

Adding Images

```
body
{
    background-image: url(/images/backgrounds/penguins.jpg);
    background-position: 0px 0px;
    background-attachment: fixed;
    background-repeat: no-repeat;
    background-color: white;
    margin: 0px;
}
```

- Image also very important
- Define the fix size before upload it (need to prepare the images)



Optimizing Images

- You need to minimize the file size of images to be used on your site while retaining as much image quality and color integrity as possible.
- Best format for web (.png,.gif) then .jpg
- Disabling Image Download
 - Disable the right-click (and therefore Save Image) feature
 - Place invisible layers over your images
 - Store images in a protected directory

```
<div id="L2" style="position:absolute; z-index:10">  
    
</div>  
  <div id="L1" style="position:absolute; z-index:5">  
      
  </div>
```

- Z-index is 3th dimension on webpage. Higher number will cover the below one

Slicing Images for Layout

- One popular technique for creating Web layouts is slicing
- In slicing, you design the layout as a graphic file
- Use Ur preference tool
 - Design the layout in the graphics editor,
 - Cut the layout into divisions or slices based on the different sections of the page.
 - Export the slices as individual graphics files, you also get an HTML table that reassembles the pieces in a Web browser.

Navigation

- Removing Underlines

```
a:link, a:visited {  
  text-decoration: none;  
}
```

- Add the others

```
a:link, a:visited {  
  text-decoration: none;  
  font-weight: bold;  
  color: #999;  
  background-color: #666;  
}
```

Changing Cursors

- Change the cursor when the mouse pointer rolls over a link

```
a:link, a:visited {  
  cursor: move;  
}
```

- Rollovers

```
a:link {  
  color: #777;  
  text-decoration: none;  
}  
a:visited {  
  color: #333;  
  text-decoration: none;  
}  
a:link:hover, a:visited:hover {  
  color: #777;  
  background-color: #ccc;  
}  
a:link:active, a:visited:active {  
  color: #ccc;  
  background-color: #ccc;  
}
```

```
div.satu:link{  
  background-color: blue;  
  width: 300px;  
  height: 80px;text-decoration: none;  
}
```

```
div.satu:hover{  
  background-color: olive;  
  width: 300px;  
  height: 80px;text-decoration: none;
```


Nongraphical Rollovers

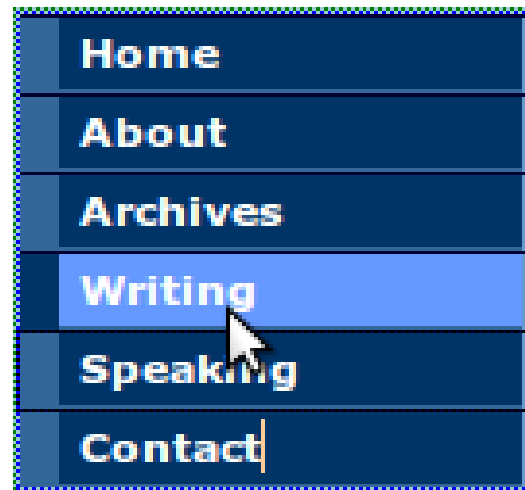
- First, mark up the list of links in an unordered list so that they wrap around a div element with an id attribute

```
<div id="nav">
  <p>Ex of navigation:</p>
  <ul>
    <li><a href="...">Home</a></li>
    <li><a href="...">About</a></li>
    <li><a href="...">Archives</a></li>
    <li><a href="...">Writing</a></li>
    <li><a href="...">Speaking</a></li>
    <li><a href="...">Contact</a></li>
  </ul>
</div>
```

- Next, use the border property on the anchor elements to create the bulk of the design

```
#nav p {
  display: none;
}
#nav {
  font-family: Verdana, Helvetica, Arial, sans-serif;
  font-size: 0.7em;
  font-weight: bold;
  width: 12em;
  border-right: 1px solid #666;
  padding: 0;
  margin-bottom: 1em;
  background-color: #9cc;
  color: #333;
}
#nav ul {
  list-style: none;
  margin: 0;
  padding: 0;
}
#nav ul li {
  margin: 0;
  border-top: 1px solid #003;
}
```

```
#nav ul li a {
  display: block;
  padding: 2px 2px 2px 0.5em;
  border-left: 10px solid #369;
  border-right: 1px solid #69c;
  border-bottom: 1px solid #369;
  background-color: #036;
  color: #fff;
  text-decoration: none;
  width: 100%;
}
html>body #nav ul li a {
  width: auto;
}
#nav ul li a:hover {
  border-left: 10px solid #036;
  border-right: 1px solid #69c;
  border-bottom: 1px solid #369;
  background-color: #69f;
  color: #fff;
}
```



Horizontal Menus

- Constructed set of unordered links, as previous slide
- Set the CSS rules for the navigation structure

```
#navsite h5 {  
  display: none;  
}  
#navsite ul {  
  padding: 3px 0;  
  margin-left: 0;  
  border-bottom: 1px solid #778;  
  font: bold 12px Verdana, sans-serif;  
}  
#navsite ul li {  
  list-style: none;  
  margin: 0;  
  display: inline;  
}  
#navsite ul li a {  
  padding: 3px 0.5em;  
  margin-left: 3px;  
  border: 1px solid #778;  
  border-bottom: none;  
  background: #DDE;  
  text-decoration: none;  
}
```

```
#navsite ul li a:link {  
  color: #448;  
}  
#navsite ul li a:link {  
  color: #448;  
}  
#navsite ul li a:visited {  
  color: #667;  
}  
#navsite ul li a:link:hover, #navsite ul li a:visited:hover {  
  color: #000;  
  background: #AAE;  
  border-color: #227;  
}  
#navsite ul li a#current {  
  background: white;  
  border-bottom: 1px solid white;  
}
```

