

Gabdulla Tuqay: A Shining Star of Tatar Literature

- Preparing Your Page Layout
- Starting Your Web Site
- Adding a CSS file
- Adding Hyperlinks
- Adding Information to the Page
 - Using CSS Classes
 - Inline Styling
 - Adding Pages
- Adding Photo Gallery

Introduction

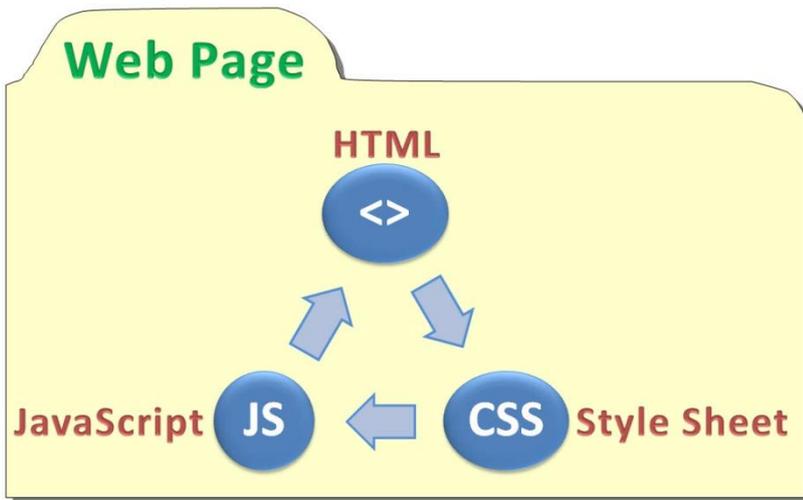
Oh, my native tongue, there is no other
like the one spoken by my father and mother.
Whatever I have learned since when I was young
It is all thanks to you, my sweet native tongue.

Gabdulla Tuqay (1909)

Gabdulla Tuqay (1886 - 1913) was a Tatar poet, a classic of the Tatar literature, a critic and a publisher. He is often referred as the founder of the modern Tatar literature and the modern Tatar literary language, which replaced Old Tatar language in literature. In this project, you are going to make a Web site about Gabdulla Tuqay to present his biography, literary personality and some of his poems.

A **Web site** consists of **Web pages**. Each Web site contains a **home page**, which is the first document users see when they enter the site. Each page contains text and images. All pages are connected to each other through the **hyperlinks**. Thus, Web pages are also called **hypertext documents**. Web browsers typically display hyperlinks using different colored text and underlining.

The Web pages operates on three technologies: HTML, CSS and JavaScript. Your Web Browser knows how to interpret them. The basic language of Web is HTML. HTML refers to **Hypertext Markup Language**. **HTML** can show your text and images, but to place them in the correct location on the page and to give formatting to the page elements, you need **CSS**. CSS refers to **Cascading Style Sheet**. **JavaScript** tells the browser how to change the web page in response to user actions and other events. In brief, HTML is responsible for the content, CSS is responsible for the design and JavaScript is responsible for the interactivity.



Picture X.1: The three fundamental Web page technologies.

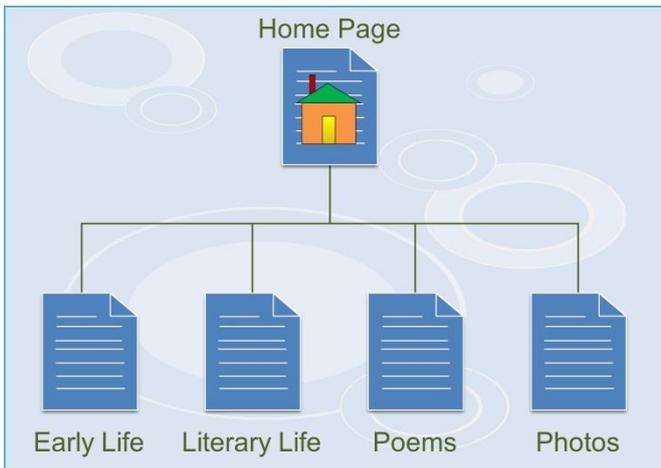
HTML, CSS and JavaScript codes are written in plain text format. You can write them in any text editor program and save in appropriate extension: **.htm** or **.html** files, **.css** for style sheet files, and **.js** for JavaScript files. However special Web development software will help you for building Web sites. Microsoft WebMatrix, Adobe DreamWeaver, Microsoft Visual Studio, Microsoft Expression Web, HTML-Kit Tools, and CoffeeCup Visual Site Designer are some of the Web page design and development tools. We are going to use Microsoft WebMatrix to develop our Web site in this project. You can also find online Web development tools on the Internet to develop and publish your sites.

The **Microsoft WebMatrix** program will be used for creating the Web site in this project. WebMatrix is a free and lightweight Web development tool. You can create, publish, and maintain your Web site with ease. If you want, you can also use Notepad program, but WebMatrix has the intellisense, it means that when you start writing the codes, it shows the possible codes which you can add.

Step 1: Making Site Diagram

Site diagram (also called site map) is the visual map of the pages in a Web site. It is a good practice to prepare the site diagram of a Web site before starting the actual building. Since every Web site has a home page (index page), the center or starting of a site diagram is always the home page.

You are going to build a simple Web site in this project. The site is consist of the home page and four additional pages: Early life, literary life, poems, and photo gallery. The site diagram of the site will be like in Picture X.2.



Picture X.2: Web site diagram.

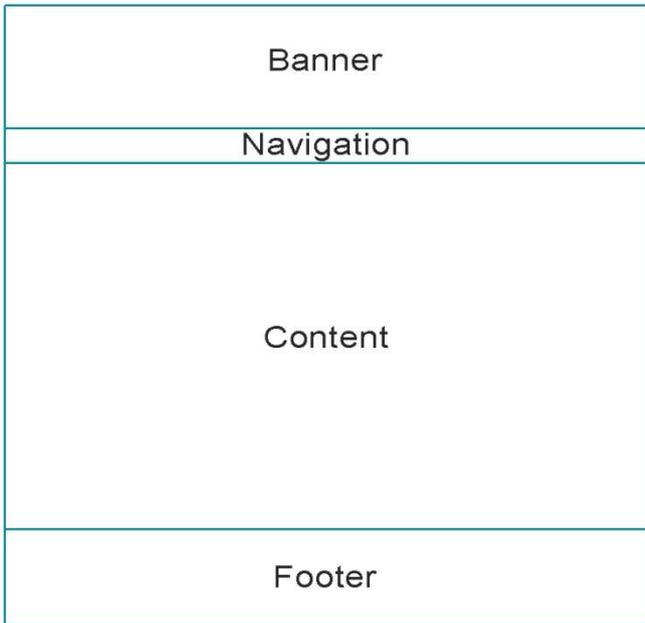
Practice: Web Site Diagram

Open a Web site on the Internet (for example your school site) and draw its site diagram. You can draw it with a pencil on paper or you can use a drawing program.

Step 2: Preparing Page Layout

A page layout is the design of your Web page. Using a **wireframe** drawing is a common way to illustrate the page layout. It is a simple visual guide to show you what a Web page would look like, without using any graphics or text. A wireframe may contain the following page elements: logo, banner, navigation, content, footer and whitespace.

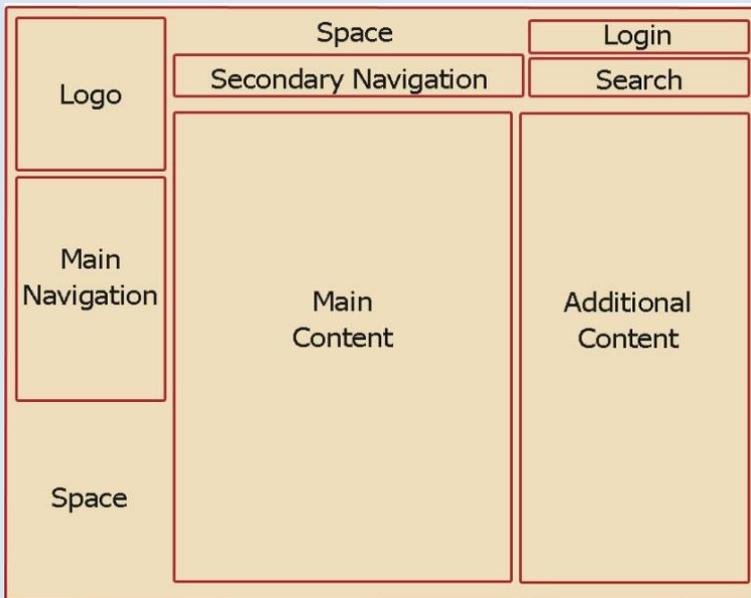
Before drawing the wireframe of your Web page, you have to decide the parts of your page. Our Web page will contain a banner, main navigation, content, and footer parts. There is an unlimited amount of ways to place them on a page. We are going to place out page element like in the picture X.3. Some parts will be common in all pages. The banner, navigation and footer will be the same in all pages. Only content will be different. You must decide the width of the page (here 900 pixels) and the general appearance of each part.



Picture X.3: Wireframe of the page.

Practice: Wireframing

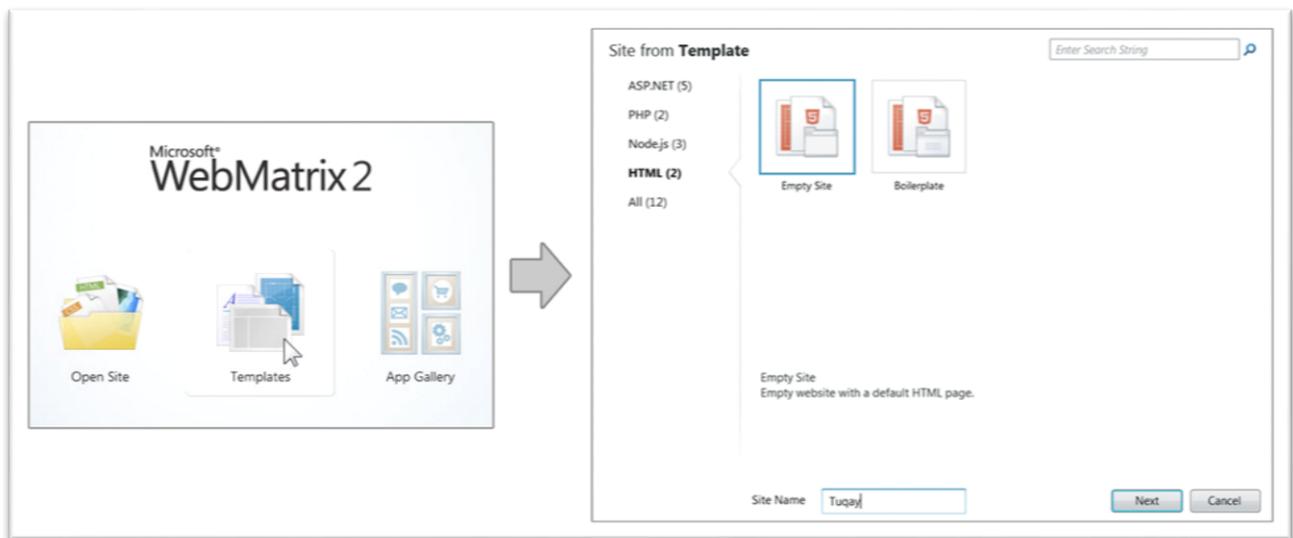
1. Generate some alternatives page layouts for our project.
2. Open several Web sites on the Internet and draw their wireframes.



Picture X.4: The wireframe of the site Wikipedia.

Step 3: Starting Your Web Site Project

- Start the WebMatrix, and choose [Templates](#).
- Click [HTML](#) and choose [Empty Site](#).
- Type the site name (**Tugay**) in the [Site Name](#) text box and click [Next](#) (Picture X.5).



Picture X.5: Starting the Web site project.

An empty Web site is created with an **index.html** file. An index (default) page is the first page that runs in a web site often called **Home page**.

- Double click the index.html file to see its content (Picture X.6).

```

<html lang="en">
  <head>
    <meta charset="utf-8" />
    <title></title>
  </head>
  <body>

  </body>
</html>

```

Picture X.6: The structure of an HTML Web page.

An HTML page contains HTML tags and plain text. HTML tags are keywords surrounded by angle brackets like **<html>**. An html page starts and finishes with **<html>...</html>** tags. There are two sections in an html page: **head** and **body**. In the head section you write information about the page while the body section contains the information that will be shown to the visitors. **UTF-8** defines what character coding is used in the page.

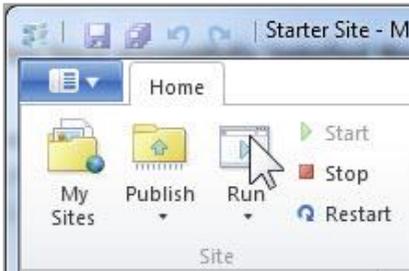
UTF-8 is a variable width Unicode format that is compatible with ASCII or plain text for the basic alphanumeric characters. By using the 'upper half' of the 8 bit ASCII set and extension codes, it can handle over a million unique characters. It's a requirement that any Web page defines the formatting of the page within its header info, so any browser knows exactly how to interpret the content.

- Type the title of the page between the title tags (Picture X.7).

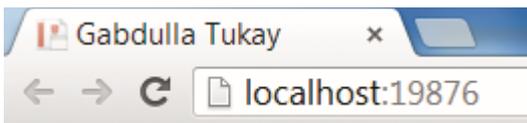
```
<head>
  <meta charset="utf-8" />
  <title>Gabdulla Tukay</title>
</head>
```

Picture X.7: Adding the title of the page.

- Click the [Save](#) and then click [Run](#) to see the page in a browser. The title is shown at the title of the browser page (Picture X.8 and Picture X.9).



Picture X.8: Running the Web site.



Picture X.9: Title of the Web site.

Step 4: Dividing the Page into Sections

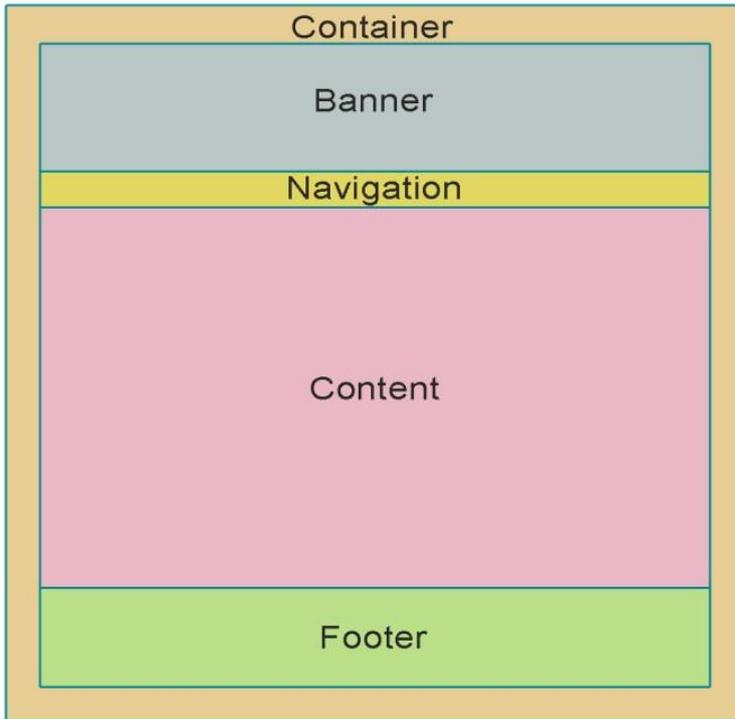
The HTML `<div>` and `<section>` tags define a division or a section in an HTML document. They are used to group block-elements to format them with CSS.

The page in our project has four parts: Banner, navigation, content, and footer. You should define a separate division for each part and an additional outer container to wrap them all (Picture X.10 and Picture X.11). Use `<div>` tags to define the divisions and give an id to each division to address them in CSS and JavaScript.

- Add the following code to the body.

```
<body>
  <div id="container">
    <div id="header"> </div>
    <div id="navigation"> </div>
    <div id="content"> </div>
    <div id="footer"> </div>
  </div>
</body>
```

Picture X.10: The divisions of the Web page.

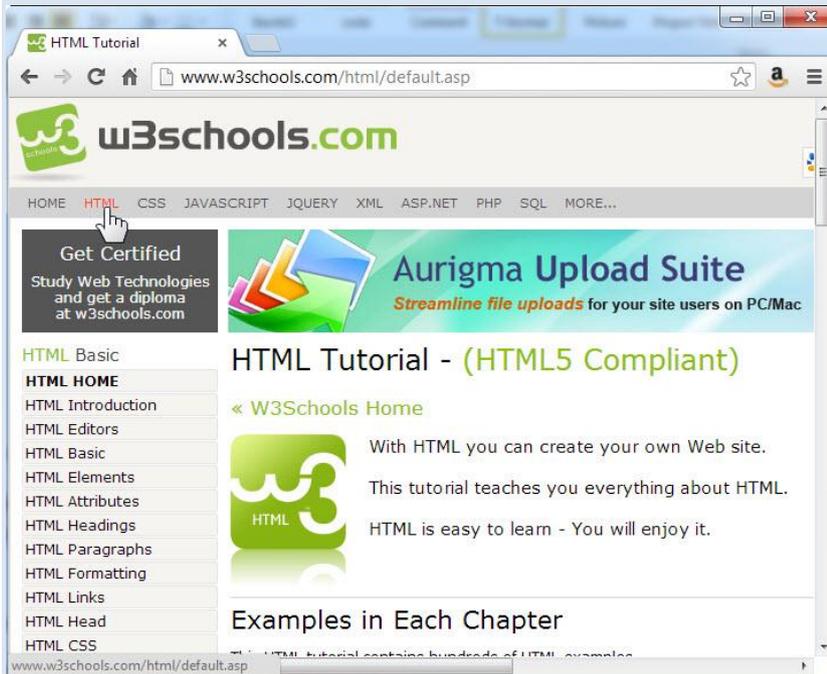


Picture X.11: The visual presentation of the HTML divisions.

IntelliSense: When you start typing the codes, a list pops up offering suggestions. It makes you more productive. After selecting your code, you can use the **Enter** or **Tab** key to commit it to your page.

Practice: HTML

Visit the site <http://www.w3schools.com/html/default.asp> and other similar sites to learn HTML basics (Picture X.12).



Picture X.12: An online HTML tutorial.

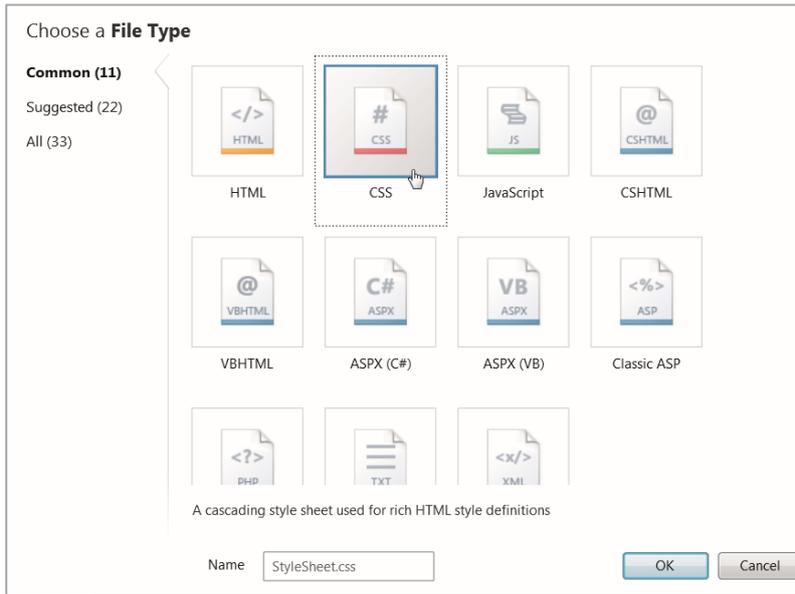
Step 5: Adding a CSS File

You have just created the divisions. It is time to apply formatting (width, color, font...) to each division. All page elements can be controlled using the CSS (Cascading Style Sheet). You need to create a CSS file and create a link to that file from your page.

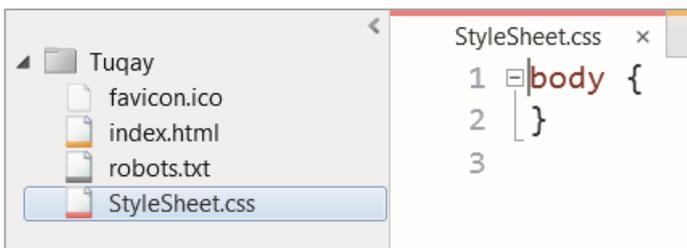
- Click **New** and **New File** (Picture X.13).
- Select **CSS** and type a name or leave it as default and click **OK**. The file is created and added to your site (Picture X.14 and Picture X.15).



Picture X.13: Adding a new file to the project.



Picture X.14: Creatig a CSS file.



Picture X.15: A CSS file with an empty body rule.

- Open the index file and add a **<link>** tag in the **head** section to link the CSS file to the index file (Picture X.16).

```

<head>
  <meta charset="utf-8" />
  <title>Gabdulla Tukay</title>
  <link href="StyleSheet.css" rel="stylesheet" type="text/css" />
</head>

```

Picture X.16: Linking the CSS style sheet file to the Web page.

- Open the CSS file and change background color of the page (Picture X.17). The CSS rule will be applied to the **body** tag of the index page.

```

body {
  background-color: #bfbbbb;
}

```

Picture X.17: Changing the background color of the Web page.

- Change the divisions formatting. The container division width should be **900** pixels and should be put at the center of the browser window automatically (Picture X.18). The CSS rule will be applied to the HTML element whose **id** is **container**.

```
#container{  
  width: 900px;  
  margin: auto;  
}
```

Picture X.18: Setting the width of the Web page.

Step 6: Adding a Banner

Prepare a banner image using a graphic editor program (Picture X.19). The dimensions of the banner are **900 x 175 px**. Create a new folder (**Images**) in the project folder and copy the banner to this folder.



Picture X.19: The banner of the site.

- Use an **** HTML tag to place the banner in the **header** division (Picture X.20). The **** tag is used to show an image on the page. Write the path and picture name in the **src** (source) attribute, and alternative text in the **alt** attribute. The **alt** attribute specifies alternate text to be displayed if for some reason the browser cannot find the image, or if the user has disabled the image files in Web browser settings. Since text-only browsers are not capable of displaying images, they greatly depend on the alt attribute.

```
<div id="header">  
    
</div>
```

Picture X.20: Adding the banner to the Web page.

- Open the CSS file, and specify the height of the header (Picture X.21).

```
#header{  
  height: 175px;  
}
```

Picture X.21: Setting the height of the header division.

- Save and run the page to see the result (Picture X.22).



Picture X.22: The banner of the site in the browser window.

Step 7: Adding Hyperlinks

- Add hyperlinks to the page using HTML `<a>` (anchor) tags (Picture X.23). A **hyperlink** is used to connect the pages to each other. You can jump from one page to another using hyperlink. The `<a>` tag is used to create hyperlinks. Write the page path and name in the **href** attribute, and the hyperlink text before the closing tag ``.

```
<div id="navigation">
  <a href="index.html">Home</a>
  <a href="EarlyLife.html">Early Life</a>
  <a href="LiteraryLife.html">Literary Life</a>
  <a href="SomeWorks.html">Some Works</a>
  <a href="PhotoGallery.html">Photo Gallery</a>
</div>
```

Picture X.23: Adding the hyperlinks to the Web page.

- Increase the space between the hyperlinks to have a good appearance by changing the hyperlink texts as in Picture X.24.

```
<a href="index.html">~ Home ~</a>
<a href="EarlyLife.html">~ Early Life ~</a>
<a href="LiteraryLife.html">~ Literary Life ~</a>
<a href="SomeWorks.html">~ Some Works ~</a>
<a href="PhotoGallery.html">~ Photo Gallery ~</a>
```

Picture X.24: Editing the hyperlink text.

- Change the formatting of the navigation and hyperlinks in the CSS file (Picture X.25 and Picture X.26).

```
#navigation{
  background-color: #eb9b74;
  text-align: center; /*set the horizontal alignment of a text to center*/
  padding: 10px; /*the space between the text and the borders*/
  border: 1px solid #6a2d26; /*draw a 1px thick solid style line*/
  font-size: large;
}
```

Picture X.25: Formatting the navigation division.

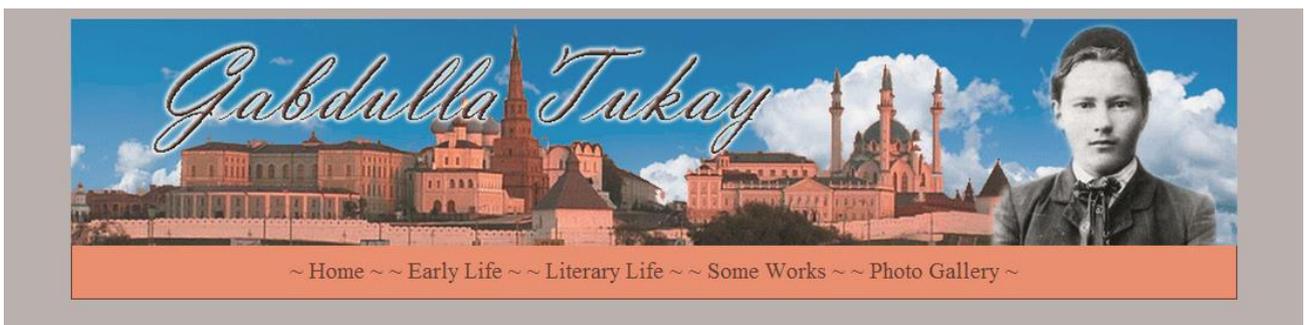
```

/*Change the color of the hyperlink and
remove the standard underline of the hyperlinks*/
a{
  color: #6a2d26;
  text-decoration: none;
}
/*Change the background of the hyperlink when
mouse over event occurs.
*/
a:hover{
  background-color: #f4f1f1;
}
/*Change the text color of the hyperlinks that
are already visited.
*/
a:visited{
  color: #000;
}

```

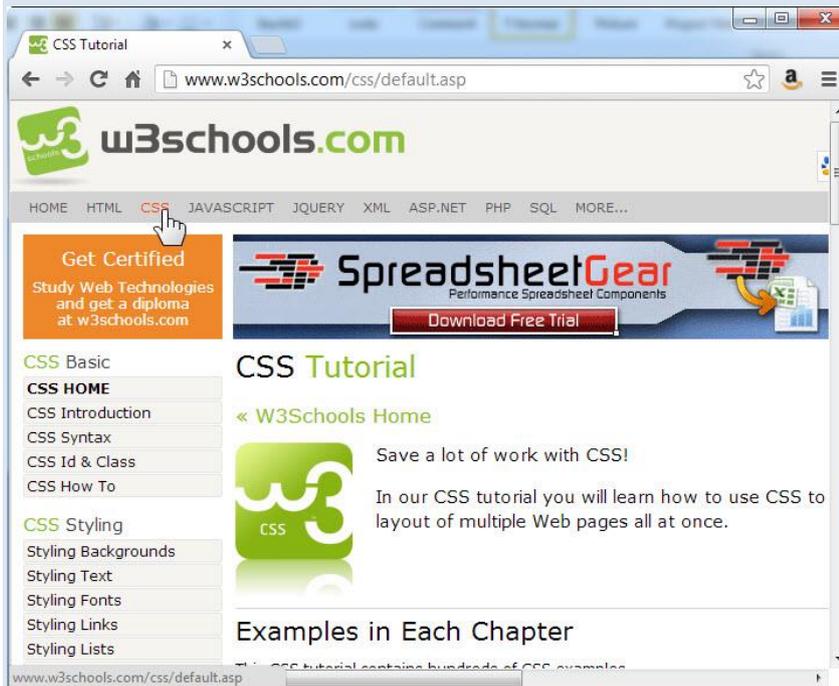
Picture X.26: Formatting the hyperlinks.

- Save and run the page to see the result (Picture X.27).



Picture X.27: The banner and the hyperlinks.

Visit the site <http://www.w3schools.com/css/default.asp> and similar sites to learn CSS basics (Picture X.28).



Picture X. 28: AN online CSS tutorial.

Step 8: Adding Information to the Content Division

- Add your text in the **content** division of the **index.html** page. In the content division, type your text and insert your images that you want to show on the home page (Picture X.29). When you write text, you can use some formatting tags:
 - `<p>...</p>` (paragraph): It makes a paragraph.
 - `...`: It makes important text.
 - `...` (emphasize): It makes emphasized text.
 - `
` (break): It inserts a single line break. It does not require closing tag.
- Add the image **Tuqay.jpg** in the content division after the first paragraph (Picture X.29). First copy your image to the **Images** folder, and use the **img** tag to show the images on the page.

```
<div id="content">
  <p>Ġabdulla Tuqay (1886 - 1913) was a Tatar poet, a classic of the Tatar
  literature, a critic and a publisher. Tuqay is often referred as the founder of
  the modern Tatar literature and the modern Tatar literary language, which replaced
  Old Tatar language in literature.
  </p>
  
  <p> <br/> </p>
  <strong>
  Ġabdulla Tuqay <br/>
  Ġабдулла Тукай <br/>
  </strong>
  <br/>
  <strong>Born:</strong> <em>26 April 1886, Quşlawıç, Kazan Governorate,
  Russian Empire</em>
  <br/>
```

```
<strong>Died:</strong> <em>15 April 1913 (aged 26), Kazan, Russian Empire</em> <br/>
<strong>Pen name:</strong> <em>Şüräle</em> <br/>
<strong>Occupation:</strong> <em>poet, publicist</em> <br/>
<strong>Nationality:</strong> <em>Tatar</em> <br/>
<strong>Genres:</strong> <em>romantism, realism</em> <br/>
<strong>Influences:</strong> <em>Alexander Pushkin, Mikhail Lermontov</em>
<br/>
</div>
```

Picture X.27: Adding content to the Web page.

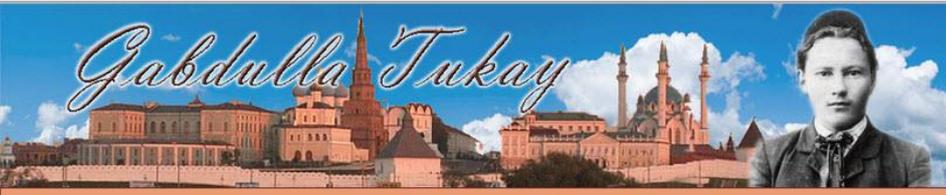
- Change the formatting of the **content** division in the CSS file (Picture X.30).

```
#content{
  background-color: #f4f1f1;
  padding: 20px;
  overflow: hidden;
}
```

Picture X.30: Formatting the content division.

overflow: hidden; One of the popular uses of setting overflow is float clearing. This means that the element with overflow applied (auto or hidden), will extend as large as it needs to encompass child elements inside that are floated (instead of collapsing), assuming that the height isn't declared.

- Save and run the page to see the result (Picture X.31).



~ Home ~ ~ Early Life ~ ~ Literary Life ~ ~ Some Poems ~ ~ Photo Gallery ~

Gabdulla Tuqay (1886 - 1913) was a Tatar poet, a classic of the Tatar literature, a critic and a publisher. Tuqay is often referred as the founder of the modern Tatar literature and the modern Tatar literary language, which replaced Old Tatar language in literature.



Gabdulla Tuqay
Габдулла Тукай

Born: 26 April 1886, Quşlawıç, Kazan Governorate, Russian Empire
Died: 15 April 1913 (aged 26), Kazan, Russian Empire
Pen name: Şüräle
Occupation: poet, publicist
Nationality: Tatar
Genres: romanticism, realism
Influences: Alexander Pushkin, Mikhail Lermontov

Picture X.31: The banne, hyperlinks and content.

Step 9: Formatting the Content Division

After adding the information into the Web page, you have to format it with CSS rules. The CSS rules are written in CSS selectors. There are three common types of CSS selectors (type, ID, and class) and there are three ways to combine CSS rules with HTML code (external, internal, and inline).

CSS Selector Types

Type Selector: It is an HTML tag selector. It corresponds with any HTML element type (<body>, <p>, <h1> etc.). The CSS rules in Picture X.30 are applied to all <p> elements in the page.

```
p{
  font-size: large;
  color: navy;
}
```

ID Selectors: The CSS rules in ID selector are applied to an element that has a specific ID attribute value. Since ID attributes must have unique values, an ID selector can never match more than one element in a Web paget. An ID selector is a name preceded by a pound character (#). The CSS rules in Picture X.31 are applied to the HTML element its ID is **comment**.

```
#comment{
  border: 1px solid blue;
  margin-top: 10px;
  margin-bottom: 5px;
  padding: 5px;
}
```

Class Selector: Class selectors can be used to select any HTML element that has a **class** attribute, regardless of their position in the HTML page. A class selector is a name preceded by a full stop (.). The CSS rules in Picture X.32 are applied to all HTML elements whose IDs is **comment**.

```
.boldblue{
  font-weight: bolder;
  color: blue;
}
```

Three Ways to Insert CSS

External style sheet: It is an external CSS file that contains only CSS code and is saved with a **.css** file extension. An external style sheet is ideal when the style is applied to many pages. Each page must link to the style sheet using the **<link>** tag in the head section of the page.

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

Internal Style Sheet: Internal style sheets are used to define styles for an HTML page. You define internal styles in the head section of an HTML page, by using the **<style>** tag. Internal CSS has the higher priority than external CSS.

```
<head>
<style>
  body {
    background-color: red;
  }
  #container{
    width: 900px;
    margin: auto;
  }
</style>
</head>
```

Inline style: It is used to apply CSS styles to one HTML element using the **style** attribute. Inline CSS has the highest priority out of the three ways you can use CSS: external, internal, and inline.

```

```

Formating the Image

We want to lace the image to the left side and move the following text to the right of the image. HTML is used to structure content and place the content one after another. To control the layout of the element and format you need CSS. We will use a CSS class selector for this task.

- Assign a class attribute (**img-left**) to the image Tuqay.jpg in the index.html file (Picture X.32).

```

```

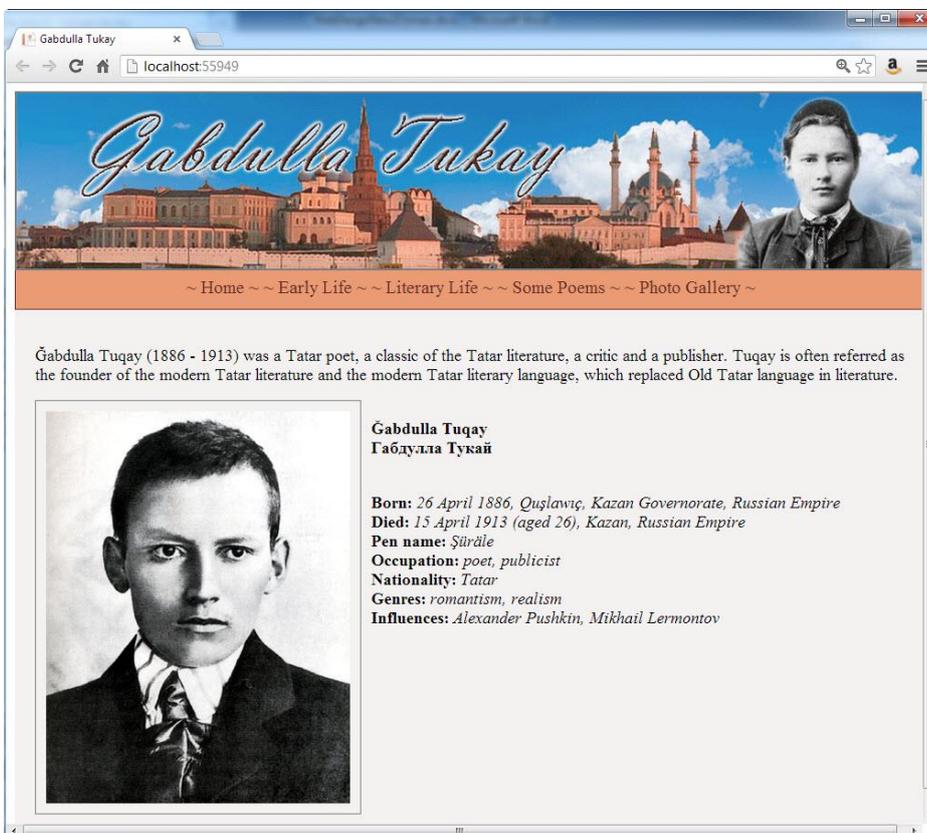
Picture X.32: Adding a class attribute.

- Define a CSC class selector for the **img-left** class in the StyleSheet.css file (Picture X.33).

```
.img-left{  
    float: left; /*push the element to the left and allow other elements to wrap  
around it. */  
    padding: 10px;  
    border: 1px solid #808080;  
    margin-right: 10px;  
}
```

Picture X.33: CSS rules of the .img-left class selector.

- Save and run the page to see the result (Picture X.34). The image is floated to the left. **Padding** gives some space between the border and text. **Margin** gives the space between the objects.



Picture X.34: After formatting the Tugay.jpg image.

Enlarging the Text

- Increase the text size of **Ğabdulla Tuqay** and **Габдулла Тукай**. You are going to change the formatting of a specific element (<p>), thus you can use inline CSS style (Picture X.35).

```
<p style="font-size: large;">
  <strong>
    Ğabdulla Tuqay <br/>
    Габдулла Тукай <br/>
  </strong>
</p>
```

Picture X.35: Using inline CSS style to enlarge the text.

- Save and run the page to see the result.

Step 10: Adding Information to the Footer Division

Adding the Information

- Add the HTML code in the Picture X.36 to the index.html page at the end of the **body** section inside the **container** division. Here **** (unordered list) tag is used to create unordered lists. **** tag is used to create list items.

```
    <div id="footer">
      <p>This site is created for only educational aims. All information
in this site is taken from the following sites:</p>
      <ul>
        <li>www.gabdullatukay.ru/eng/</li>
        <li>en.wikipedia.org/wiki/Ğabdulla_Tuqay</li>
      </ul>
    </div>
  </div>
</body>
</html>
```

Picture X.36: The content of the footer division.

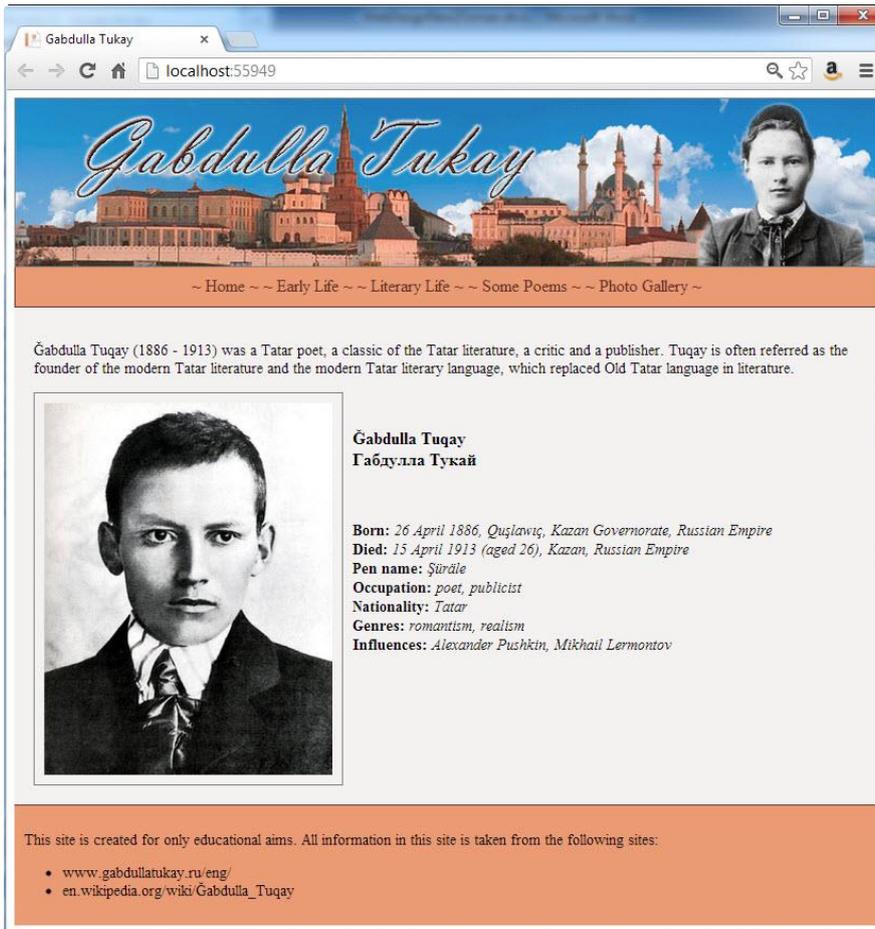
Formatting the Footer Section

- Change the formatting of the footer division in the CSS file (Picture X.37).

```
#footer{
  background-color: #eb9b74;
  padding: 10px;
  border-top: 2px solid #6a2d26;
}
```

Picture X.37: The CSS ID selector for footer.

- Save and run the page to see the result (Picture X.38).



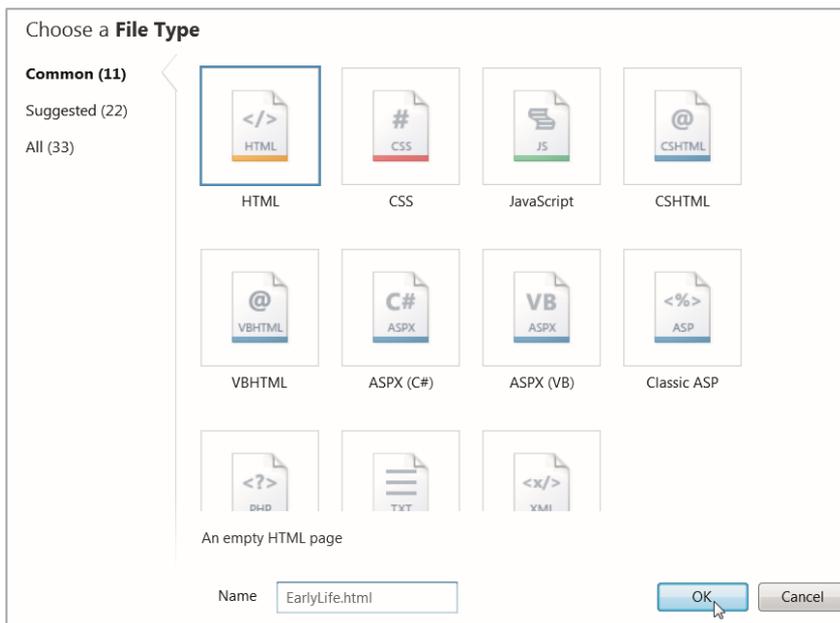
Picture X.38: After adding the footer division.

Step 11: Creating the Literary Life Page

You have completed the first page of your web site. The other pages will also have the same layout with the home page. You need to change only the information in the content division. Header, footer and navigation sections will repeat on each page. For navigation and design purposes, the sites page layout should remain consistent on different pages.

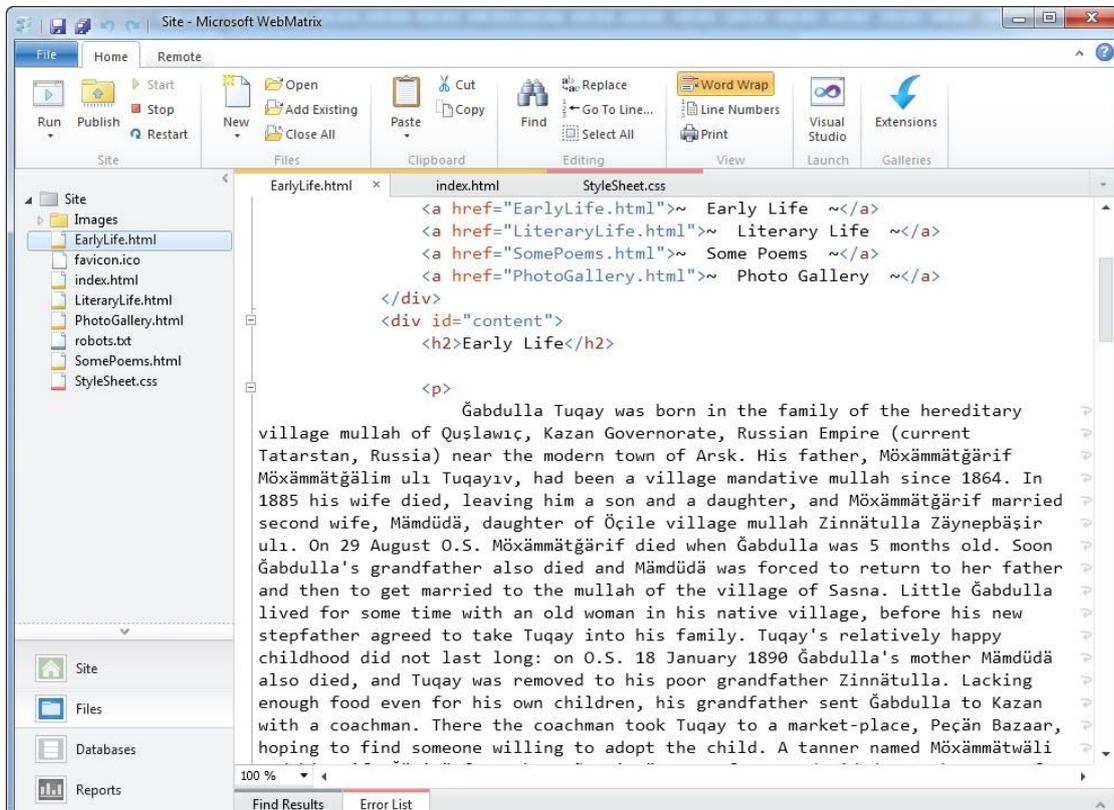
Many websites have content that's displayed on every page, like navigation block, header, and footer. To create a consistent layout along the site, those repeating content should appear on the same place of each page. Many Web sites on the Internet consist of thousands of pages. You need to use **master pages** or **templates** to create a consistent layout for medium-scale or larger Web sites.

- Create a new HTML page with the name **EarlyLife.html** (New → New File → HTML) (Picture X.39).



Picture X.39: Adding a new HTML page to the site.

- Copy all codes from the index page to the new page. Delete all codes inside the **content** division and add your new information (image X.40). You can create headings using the `<h1>...<h6>` tags. `<h1>` defines the biggest heading and `<h6>` defines the smallest heading. When you add an image, to use the **img-left** class which you created in the CSS, add class the attribute into the image tag.



Picture X.40: The content of the EarlyLife.html page.

- Format the **LiteraryLife.html** page with the CSS rules in the **StyleSheet.css** file and run it to see the result in the browser window (Picture X.41).



Picture X.41: The EarlyLife.html page on the browser.

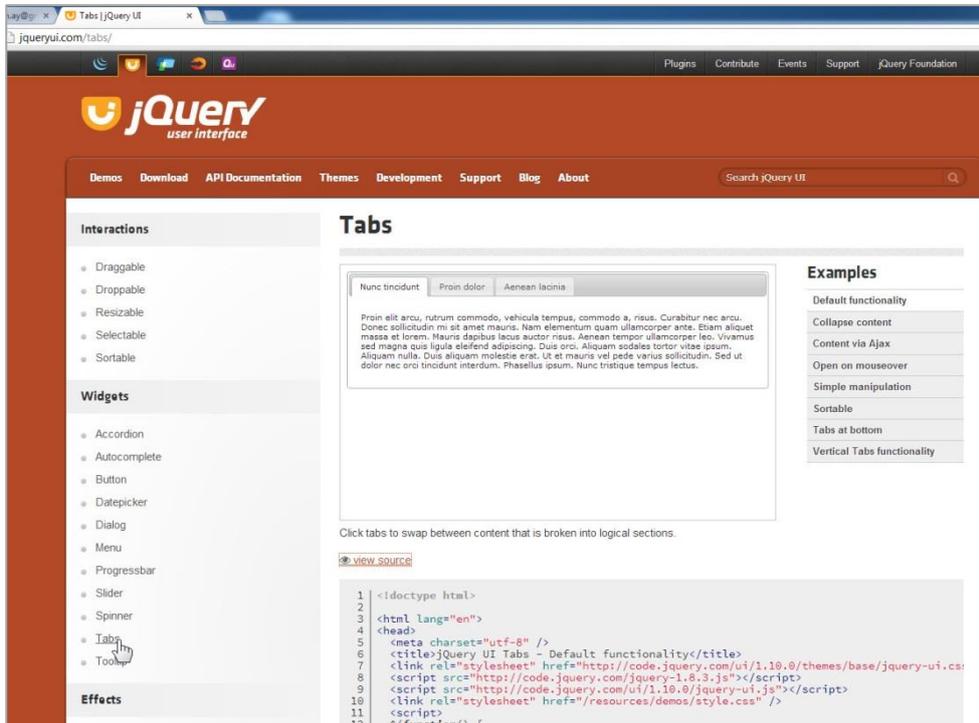
Step 12: Creating the Some Poems Page

Although Gabdulla Tukay died very young (twenty six years old), he left behind unforgettable poems. We are going to share some his poems (My Native Tongue, My Village, Shuraleh, and The Water Maid) in the page with the visitors. Instead of displaying his poems in a continuous text format one after another, we can use an element of a graphical interactive user interface like menus and tables.

Creating an interactive user interface element requires **JavaScript** programming and **CSS** designing. JavaScript programming is beyond the scope of this book. There are many open sources ready to use JavaScript libraries for the non-programmer Web developers. **JQuery** (The Write Less, Do More, JavaScript Library) is one of the common JavaScript libraries. **JQuery UI** (user interface) is a plug-in (an extension library which is built on top of JQuery library) for the JQuery library to create animations, advanced effects, widgets, and interaction mechanism.

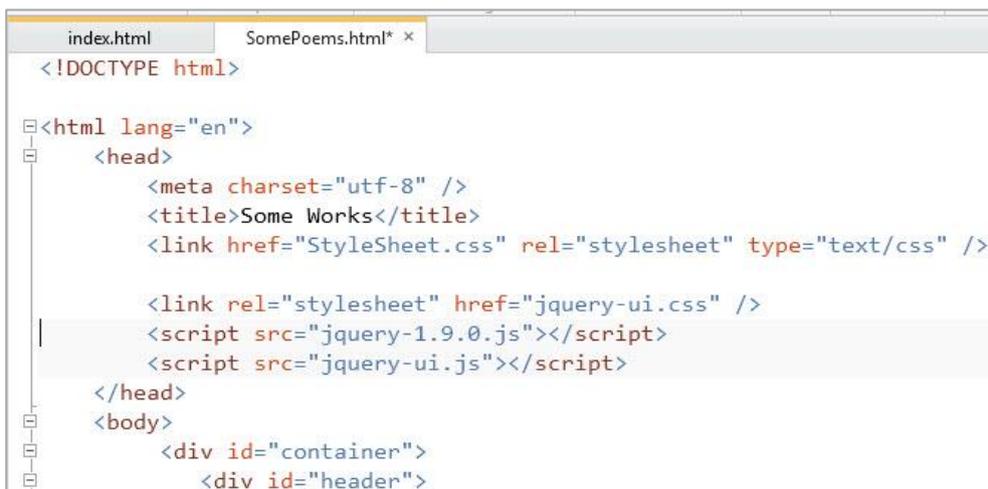
- Create a new HTML page with the name **SomePoems.html** (New → New File → HTML).

- We are going to use tabs widget to display the poems in the browser. Open the <http://jqueryui.com/tabs/> site and examine the tabs widgets to understand how to use it (Picture X.42).



Picture X.42: JQuery UI tabs widget.

- Add the JQuery, JQuery UI libraries and related CSS files (jquery-ui.css, jquery-1.9.0.js, jquery-ui.js) to use the tabs widget in the header section of the SomePoems.html file (Picture X.43). You can download the files from jQuery UI site (<http://jqueryui.com/download/>). The current versions of the files can be different in the site.



Picture X.43: Adding jQuery and JQuery UI links to the page.

- Add the corresponding JavaScript function for the tabs widget to the head section of the page (Picture X.44).

```

SomePoems.html* x
<head>
  <meta charset="utf-8" />
  <title>Some Works</title>
  <link href="StyleSheet.css" rel="stylesheet" type="text/css" />
  <link rel="stylesheet" href="jquery-ui.css" />
  <script src="jquery-1.9.0.js"></script>
  <script src="jquery-ui.js"></script>

  <script>
    $(function() {
      $( "#tabs" ).tabs();
    });
  </script>

</head>

```

Picture X.44: Adding JavaScript code to the head section for tabs widget.

- Create a div element with the **tabs** id (`<div id="tabs">`) and add an unordered list (``) of the poems in the **tabs** div in the **content** section (Picture X.45).

```

SomePoems.html* x
</head>
<body>
  <div id="container">
    <div id="header">...</div>
    <div id="navigation">...</div>
    <div id="content">
      <h2>Some Poems</h2>
      <div id="tabs">
        <ul>
          <li><a href="#tabs-1">My Native Tongue</a></li>
          <li><a href="#tabs-2">My Village</a></li>
          <li><a href="#tabs-3">Shuraleh</a></li>
          <li><a href="#tabs-4">The Water Maid</a></li>
        </ul>
      </div>
    </div>

```

Picture X.45: Unordered list of the poem names.

Add a separate div element for each poem after the unordered list of the poems. Each div should have an **id** attribute with the same **href** attribute in the unordered list (tabs-1, tabs-2, tabs-3, and tabs-4). The page should be like in Picture X.46.

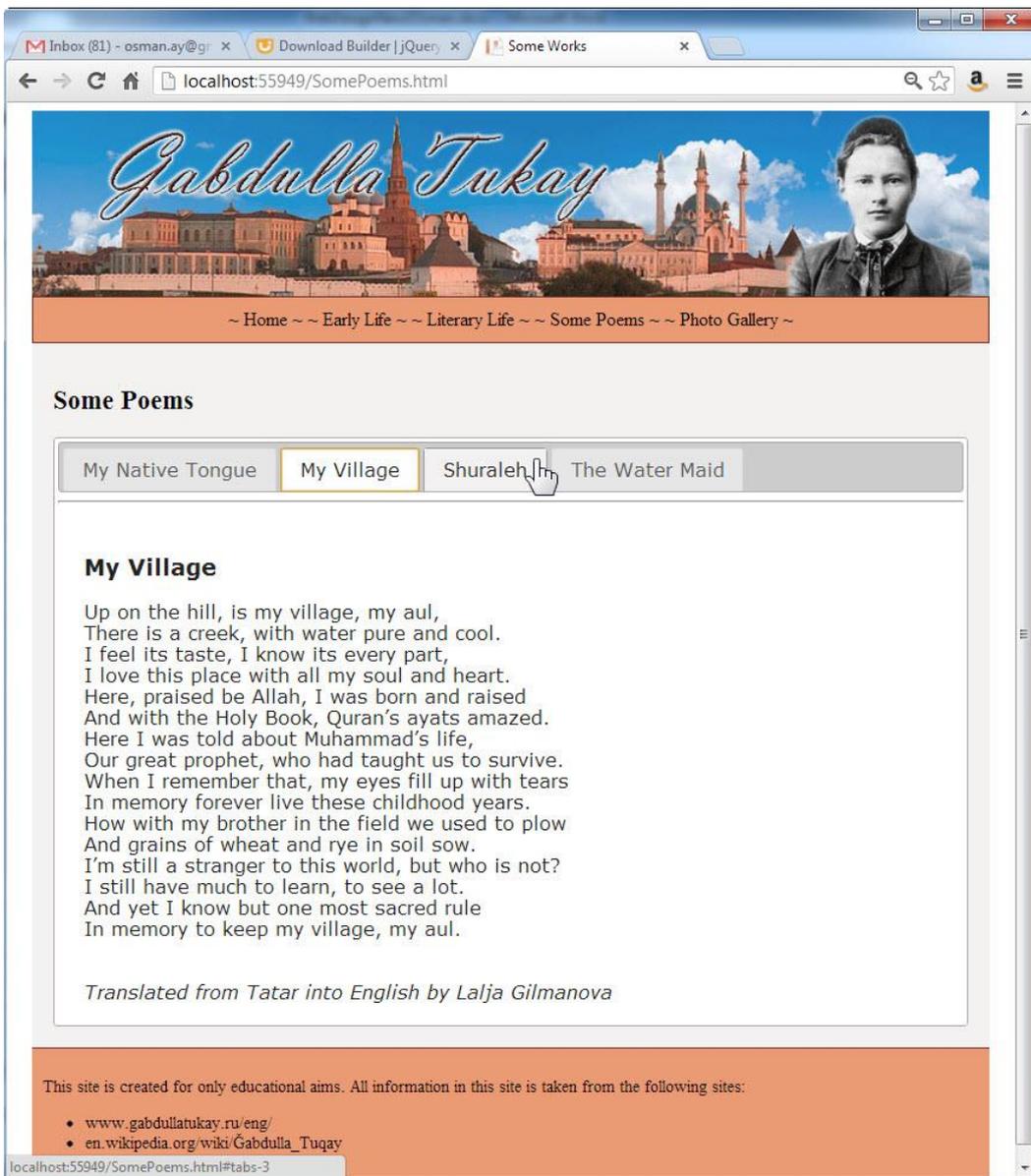
```

SomePoems.html* x
  </head>
<body>
  <div id="container">
    <div id="header">...</div>
    <div id="navigation">...</div>
    <div id="content">
      <h2>Some Poems</h2>
      <div id="tabs">
        <ul>
          <li><a href="#tabs-1">My Native Tongue</a></li>
          <li><a href="#tabs-2">My Village</a></li>
          <li><a href="#tabs-3">Shuraleh</a></li>
          <li><a href="#tabs-4">The Water Maid</a></li>
        </ul>
        <hr/>
        <div id="tabs-1">
          <h3>My Native Tongue</h3>
          Oh, my native tongue, there is no other<br/>
          Like the one spoken by my father and mother.<br/>
          .
          .
          .
          "Oh, Lord Almighty, in your mercy, please save<br/>
          My poor parents and me - I'm your humble slave."<br/>
          <br/>
          <em>Translated from Tatar into English by Lalja Gilmanova </em><br/>
        </div> <!--tabs-1-->
        <div id="tabs-2">...</div>
        <div id="tabs-3">...</div>
        <div id="tabs-4">...</div>
      </div> <!--tabs-->
    </div> <!--content-->
  <div id="footer">...</div>
</div> <!--container-->
</body>

```

Picture X.46: Adding the poems in separate div elements.

- Save and run the page to see the result (Picture X.47). Display the poems using the tabbed interface.



Picture X.47: Displaying the poems in a tabs widget.

Exercise: jQuery UI Elements

Explore the other interaction, effects, widgets, utilities, and themes in the jQuery UI site. Click the **Demos** tab and try each element to see what it does and click the **view source** link to understand how to use it in your projects.

Practice: Accordion Menu

We used tabs control to display the poems. There are some other appealing user interfaces you can use. Remove the tabs widget and use the **jQuery UI accordion** menu to display the poems (Picture X.48).

Accordion

Click headers to expand/collapse content that is broken into logical sections, much like tabs. Optionally, toggle sections open/closed on mouseover.

The underlying HTML markup is a series of headers (H3 tags) and content divs so the content is usable without JavaScript.

[view source](#)

Want to learn more about the accordion widget? Check out the [API documentation](#).

Examples
Default functionality
Collapse content
Customize icons
Fill space
No auto height
Open on hover
Sortable

Picture X.48: The accordion menu interface.

Step 13: Adding Photo Gallery

In the photo gallery page you will have the thumbnails of the images. A **thumbnail** is small version of an image. When you click the thumbnails, the original image will be opened in a new page.

- Collect your images and prepare the thumbnails of your images with the size of **200x150** pixels using photo editor software.
- Copy all images (including thumbnails) to the **Images** folder.
- Create the **PhotoGallery.html** and add the following codes in the content division to create a table with three rows and three columns (Picture X.49). The rows are created with the **<tr>** (table row) tags. In each row, the columns are created with the **<td>** (table data) tags.

```
<table>
  <tr>
    <td></td>
    <td></td>
    <td></td>
  </tr>
  <tr>
    <td></td>
    <td></td>
    <td></td>
  </tr>
  <tr>
    <td></td>
    <td></td>
    <td></td>
  </tr>
</table>
```

Picture X.49: Creating a tabel with three rows and three columns.

- Add the **thumbnail** images and hyperlinks within the **<td>** tags (Picture X.50). The hyperlinks are given not to the pages but images. Since the value of **target** attribute is **blank** (**target="_blank"**), the images will open in a new window.

```
<div id="content">
  <h2>Photo Gallery</h2>

  <table class="tformat">
    <tr>
      <td>
        <a href="Images/gallery1.jpg" target="_blank">
          </a>
        </td>
      <td>
        <a href="Images/gallery2.jpg" target="_blank">
          </a>
        </td>
      <td>
        <a href="Images/gallery3.jpg" target="_blank">
          </a>
        </td>
    </tr>
    <tr>
      <td>
        <a href="Images/gallery4.jpg" target="_blank">
          </a>
        </td>
      <td>
        <a href="Images/gallery5.jpg" target="_blank">
          </a>
        </td>
      <td>
        <a href="Images/gallery6.jpg" target="_blank">
          </a>
        </td>
    </tr>
    <tr>
      <td>
        <a href="Images/gallery7.jpg" target="_blank">
          </a>
        </td>
      <td>
        <a href="Images/gallery8.jpg" target="_blank">
          </a>
        </td>
      <td>
        <a href="Images/gallery9.jpg" target="_blank">
          </a>
        </td>
    </tr>
  </table>
</div>
```

Picture X.50: Adding the thumbnail images and their hyperlinks.

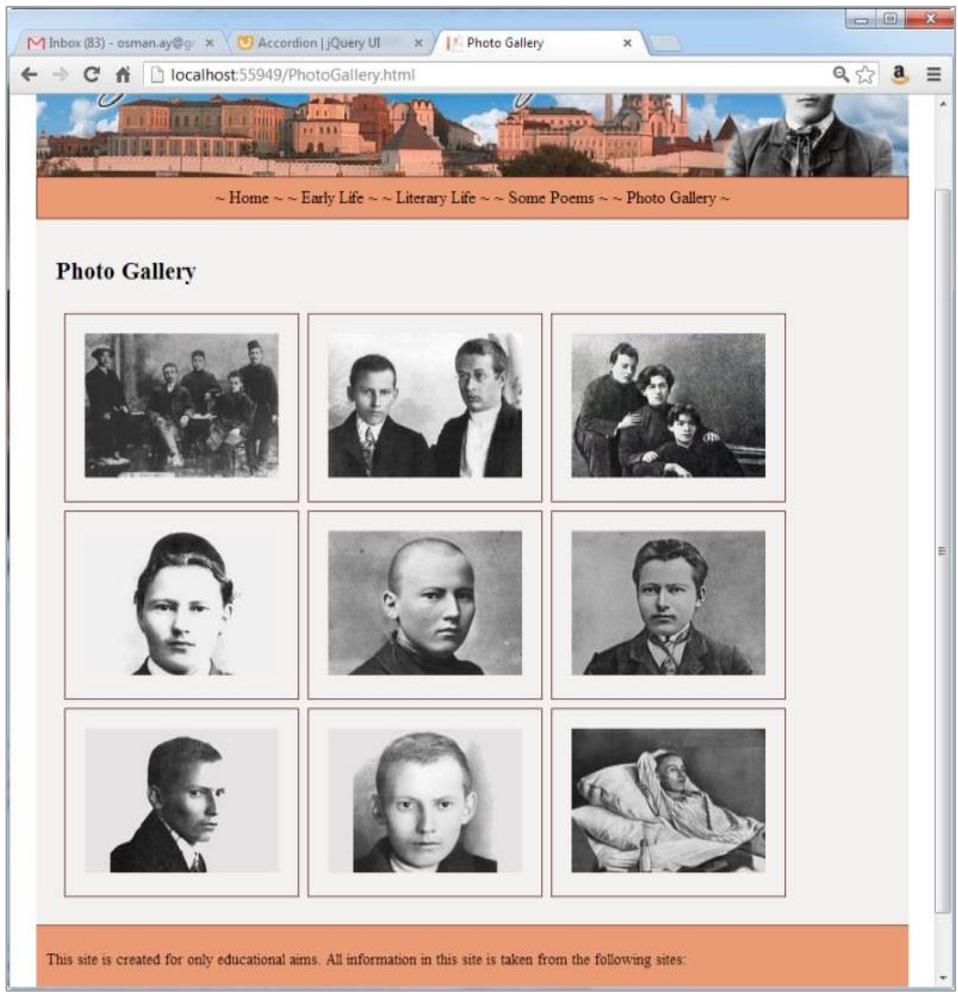
You can define a CSS class for your table.

- Add class attribute to the **<table>** tag to format it with CSS rules: `<table class="tformat">`
- Add the class selectors to the **StyleSheet.css** file to format the table (Picture X.51).

```
.tformat{  
    border-spacing:10px; /*The distance between the borders of adjacent cells.*/  
}  
.tformat td{  
    padding: 20px;  
    border: 1px solid #6a2d26;  
}
```

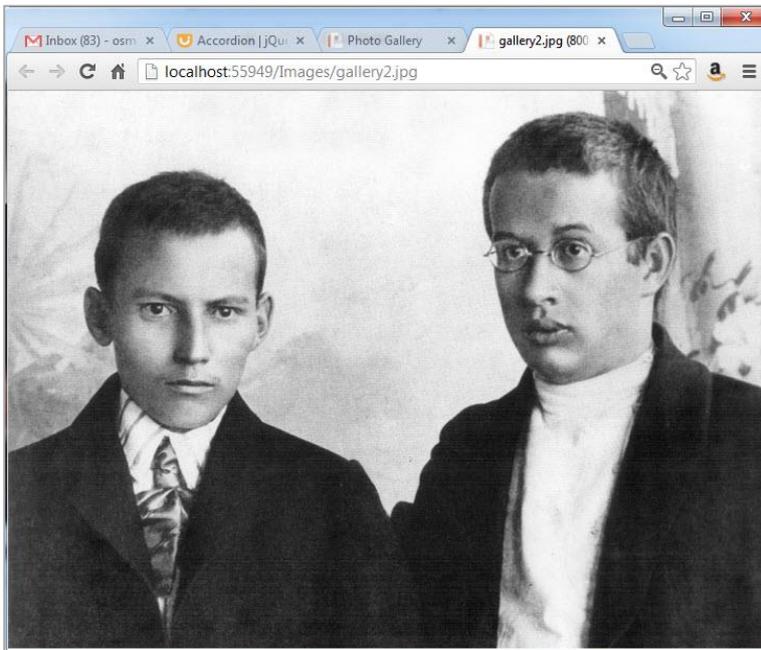
Picture X.51: Adding CSS class selectors to format the table.

- Save and run the page to see the result (Picture X.52).



Picture X.52: The photo gallery.

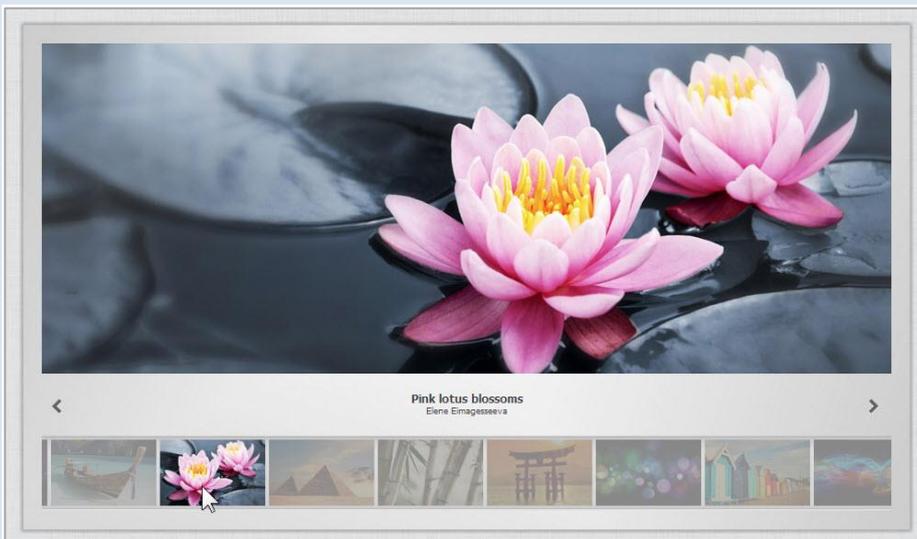
- Click any thumbnail to see that a big version of the image opens in a new window (Picture X.53).



Picture X.53: Opening full-size photo in a new browser window.

Practice: Image Gallery

There are many JavaScript and jQuery image galleries in on the Web you can download and use in your Web sites. Those galleries include slideshow, transition effects, multiple album options, CSS skinning and much more. Google the image galleries on the Web and use any of them in the **PhotoGallery.html** page to enhance usability and design of the project. There is a sample image gallery in the Picture X.54.

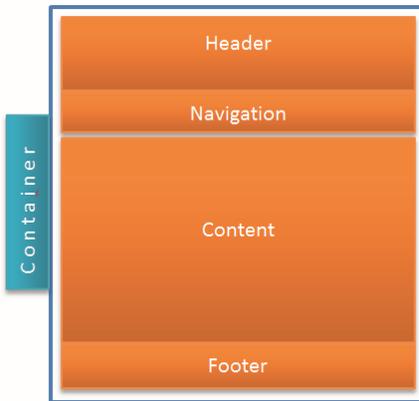


Picture X.54: A sample image gallery.

Project

Make a new web site about your favorite poet. The project will provide the following requirements:

1. A Page Layout
2. A Banner
3. A Navigation and pages
4. A CSS file
5. A JQuery UI
6. A Photo gallery



~ Home ~ ~ Early Life ~ ~ Literary Life ~ ~ Some Poems ~ ~ Photo Gallery ~

My Native Tongue
My Village
Shuraleh
The Water Maid

My Native Tongue

Oh, my native tongue, there is no other
 Like the one spoken by my father and mother.
 Whatever I have learned since when I was young
 It is all thanks to you, my sweet native tongue.
 My first words to utter I was not yet able,
 Mother sang me lullabies, while rocking my cradle.
 I'll forever remember the tales from my granny -
 She told me one each night, of them she knew so many
 If it weren't for you, oh my beautiful tongue,
 To this day I'd be speechless and hopelessly numb...
 But you are so gracious, together did we grow,
 You taught me to express my happiness and sorrow.
 Oh, my native tongue, how could I ever forget
 That first prayer to Allah, which I myself read?
 "Oh, Lord Almighty, in your mercy, please save
 My poor parents and me - I'm your humble slave."
Translated from Tatar into English by Lajla Gilmanova

```

StyleSheet.css x
1 body {
2   background-color: #bfbbbb;
3 }
4 #container{
5   width: 900px;
6   margin: auto;
7 }
8 #header{
9   height: 175px
10 }
11 #navigation{
12   background-color: #eb9b74;
13   text-align: center;
14   padding: 10px;
15   border: 1px solid #6a2d26;
16   font-size: large;
17 }
18 a{
19   color: #6a2d26;

```

Photo Gallery






Questions

1. What is HTML?
 - a. Hypertext Markup Language
 - b. Hyperlink Methods Language
 - c. Hypertext Methods Language

- d. Hyperlink Markup Language
 - e. Hypertext Master Language
2. What is CSS?
- a. Cascading Style Sheet
 - b. Connecting Style Software
 - c. C# Style Software
 - d. Connecting Style Service
 - e. Compact Spread Spectrum
3. Which main parts does an HTML page include?
- a. Head and body
 - b. Meta and title
 - c. Head and container
 - d. Meta and container
 - e. Head and meta
4. Which one of the following codes connects a CSS file to the page?
- a. `<link href="StyleSheet.css" rel="stylesheet" type="text/css" />`
 - b. `<link src="StyleSheet.css" rel="text/css" type="stylesheet" />`
 - c. `<link src="StyleSheet.css" rel="stylesheet" type="text/css" />`
 - d. `<link href="StyleSheet.css" rel="text/css" type="stylesheet" />`
 - e. `<link href="StyleSheet.css" rel="css" type="stylesheet" />`
5. Which attribute is used to write the path and picture name when you add an image?
- a. Src
 - b. Href
 - c. Link
 - d. Rel
 - e. Style
6. Which of the following is used to put space between the text and the border?
- a. Padding
 - b. Margin
 - c. Float
 - d. Border
 - e. Indent
7. Which of the following is used to change formatting for the hyperlinks when you do mouse over them?
- a. Hover
 - b. Visited
 - c. Over
 - d. Linked
 - e. On
8. Which tag moves the cursor to the next line?
- a. `
`
 - b. `<div>`
 - c. `<p>`
 - d. `<h1>`
 - e. `<hr>`
9. Which of the following shifts the object?
- a. Float

- b. Padding
- c. Margin
- d. Border
- e. Indent

10. Which tag is used to create unordered lists?

- a.
- b.
- c. <p>
- d. <a>
- e.

11. What is JQuery?

- a. A JavaScript library
- b. A JavaScript user interface
- c. A JavaScript animation
- d. A JavaScript widget
- e. A JavaScript advanced effect

```
.boldblue{  
  font-weight: bolder;  
  color: blue;  
}
```

12.

Which CSS selector type is used in this code?

- a. Class selector
- b. ID selector
- c. Type selector
- d. HTML selector
- e. Element selector

13. ``

Which way is used to insert CSS code to HTML?

- a. Inline style
- b. Internal style sheet
- c. External style sheet
- d. Embedded style sheet
- e. Image style

14. X

15. x