

END OF YEAR REVIEW

HTML CSS JAVASCRIPT PHP DATABASE

HTML

FORMATTING WEB CONTENT

BASIC TAGS

- **Tag / Explanation**

- <a> / Links to another file
- <p> / Creates a paragraph
- / Creates an image
-
 / Starts a new line
- <form> & <input> / Create a form

- **Example**

- Click Here
- <p>Hello World</p>
-
-

- <form method="post" action="process.php">
 <input type="text" name="email" />
 </form>

BASIC CONCEPTS

An **attribute** is extra information between < and > (other than the tag)

For example *href* in a link is an attribute:

```
<a href="somefile.html">Click Here</a>
```

In the following tag, *id*, *class*, *src*, and *style* are all *attributes*:

```

```

REMEMBER!

Tags start and end with square brackets: <>

Tags usually come in pairs: <p> </p>

Common tags include <a>, <p>, ,
, <form>, <input> (hint: know these well for the test)

Attributes add extra information to a tag

Attributes appear within the square brackets:

Common attributes include *src, href, class, id, style, name, method, action* (hint: know these well for the test)



CSS

STYLING WEB CONTENT



BASIC CONCEPTS

- CSS is used to add *style* to our HTML
- CSS can be *inline* or part of a *stylesheet*
- Styles in a *stylesheet* can be applied using *ids* or *classes*
- CSS allows you to control things like colors, margins, padding, font face, font style, borders, etc.
- Many items can have the same *class*
- Only one item may have the same *id*

- Inline CSS example:
`<p style="font-weight: bold; color: #ff0000"></p>`
- Stylesheet example:
 - (On the HTML page)
`<p id="myStyle" class="boldFont"></p>`
 - (In the CSS file)
`#myStyle { color: #ff0000; }
.boldFont { font-weight: bold; }`

(hint: know the difference between a *class* and an *id*)



JAVASCRIPT

FRONT-END PROGRAMMING



BASIC CONCEPTS

- Used for front-end programming
- Used to manipulate on-screen HTML
- Used to validate form data *before* it is sent to PHP for server-side processing
- Does not connect directly to database
- We were specifically using the JQuery library in this class
- JQuery allows us to target HTML elements using same *class* and *id* syntax as CSS

(hint: understand targeting using *classes* and *ids*)



PHP

SERVER-SIDE PROGRAMMING



BASIC COMMANDS

■ Command / Explanation

- echo / Sends information to the screen
- while / Loops through data
- if else / Makes a decision
- include / Adds code from another PHP file
- setcookie / Sets a cookie

■ Example

- echo "<p>hello world</p>";
- while (\$line = mysql_fetch_assoc(\$result)) { ... }
- if (\$line["user"] == \$user) { ... } else { ... }
- include("another-file.php");
- setcookie("mycookie", "mydata", time() + 3600, "/");

(hint: know these five examples well)

BASIC CONCEPTS

- Used to process server-side information
- Connects directly to database
- Variables start with dollar sign: \$variable
- Commands end with semi-colon ;
- Database Process:
 - connect to the database
 - select information from table
 - loop (while) through information
- Login Process:
 - get username / password via a form
 - check information against the database
 - if correct, set a cookie

DATABASE

STORING AND MANIPULATING INFORMATION

BASIC CONCEPTS

- Stores information for later use
- You need to connect to a server before connecting to the *database*
- A *database* contains many *tables*
- A *table* contains many *fields*
- A *table* contains many *rows* (records)
- A *database* command is called a *query* (or more specifically an *SQL query*)
- Basic commands: **SELECT, INSERT, WHERE, ***

■ **Command / Explanation**

- **SELECT** / gets information from a table
- **INSERT** / puts information into a table
- **WHERE** / limits the data we get back
- ***** / used to indicate “everything”

QUERY EXAMPLES

- SELECT Example:

```
SELECT * FROM myTable WHERE myID = "4"
```

will select all *fields* from the *table* called “myTable” where the *field* called “myID” has a value of “4”

- INSERT Example:

```
INSERT INTO myTable (myID) VALUES ("4")
```

will insert the value “4” into the *table* “myTable” under the *field* “myID”

(hint: know these examples well)

PUTTING IT ALL TOGETHER

SOME ADDITIONAL CONCEPTS

FORMS EXAMPLE

- **HTML FORM**

```
<form method="post" action="contact.php">
  <label>Email:</label>
  <div class="form-control"><input type="text" name="email" id="emailAddress" /></div>
</form>
```

- **CSS TARGET**

```
.form-control { ... }
#emailAddress { ... }
```

- **JQuery TARGET**

```
$("#emailAddress").val()
```

- **PHP FILE (contact.php)**

```
$_POST["email"]
```

(hint: know this example well)

DATABASE EXAMPLE

- **PHP**

```
include ('database-connection.php');

$query = 'SELECT * FROM myTable';
$result = mysql_query($query);
while ($line = mysql_fetch_assoc($result)) {
    echo "myData:" . $line["myData"] . "<br/>";
}
```

- The above code will use 'database-connection.php' to connect to the *database*, then select all *fields* from the *table* called "myTable". The results from the SQL query will be stored in \$result and processed using a *while* loop. Each *row* (or *record*) in the *table* will echo information to the screen.

(hint: know this example well)