```
<?php
echo “These are the contents of test2.php!”;
?>
```

The output of running test.php is:

```
test2.php outputs the following: These are the contents of test2.php!
```

This statement is useful for re-using code without writing it multiple times, for creating templates, or for separating your code into manageable files.

Transferring Data:
A critical part of making your website dynamic is the ability to accept input from users. Two standard methods of sending data to a web server are through “GET” and “POST” transfers. PHP provides access to these data through two superglobal (accessible everywhere in your program) arrays called **$_GET** and **$_POST**.

GET:
Data sent through the URL (or through other methods by external programs) is captured in the GET array. The following is a comparison between a URL with GET info, and the corresponding **$_GET** array created by PHP:

```
http://example.com/test3.php?title=CoolPage
```

There are many server side scripting languages, PHP being one of them. A good place to start learning PHP Server Side scripting is here:

http://www.w3schools.com/php/
APPENDIX CLIENT SIDE SCRIPT

Client-Side scripting (most commonly, JavaScript) allows for a dynamic and interactive user experience on your website. Using JavaScript, the elements on the page can be adjusted, created, or removed without having to refresh or navigate to a new page. All modern browsers are also able to send data to and from servers using JavaScript.

The DOM:
JavaScript’s Document Object Model (DOM) provides the interface through which we can access elements and input on web pages. Various built-in functions allow us to target, manipulate, and create/remove certain elements from the “document tree” that comprises the page.

Embedding JavaScript:
JavaScript can either live inside of the HTML page you create, or can reside in an external file which is *referenced* in HTML files where it is used.

Inline:
The *inline* method is done simply using the <script> tag which is usually placed between the <head> and </head> tags.:

```
<!DOCTYPE html>
<html>
<head>
  <meta charset=UTF-8>
  <title>JavaScript Testing</title>
  <script>
    alert("Welcome!");
  </script>
</head>
<body>
  <p>Nothing to see here!</p>
</body>
</html>
```

This script would be very annoying; it simply prompts "Welcome!" to the user upon loading the page.

External Referencing:
Alternatively, this same script could be contained in an external file, myscript.js, and referenced in our HTML page. The HTML page would look as follows:

```
<!DOCTYPE html>
<html>
<head>
  <meta charset=UTF-8>
  <title>Javascript Testing</title>
  <script src=myscript.js>
  </script>
</head>
<body>
  <p>Nothing to see here!</p>
</body>
</html>
```

Notice the "src=myscript.js" that was added to the <script> tag. This directs the browser to the file that contains the script that should be executed. If, inside of myscript.js, we had written:

```
alert("Welcome!");
```

Then this alternative method would have the same effect: *Welcome!* would be alerted on the page.

A good place to start learning JavaScript is here:

http://www.w3schools.com/js/js_intro.asp