LOYOLA UNIVERSITY CHICAGO | CENTER FOR TEXTUAL STUDIES AND DIGITAL HUMANITIES

DIGH 401 - Introduction to Computing

Fall Semester 2014

Week 9

Today's Class

- Week 7 Exercise
- Manipulating data (continued)

DIGH 401 - Introduction to Computing

Week 7 Exercise - Programming basics

```
<?php
$total = 0;
$booster1 = 50;
$booster2 = 100;

function rocketThrust()
{
global $total, $booster1, $booster2;
$total = $booster1 + $booster2;
}

rocketThrust();
echo 'Rocket thrust is currently stable at '.$total.' newtons';
?>
```

DIGH 401 - Introduction to Computing

Week 7 Exercise - Part 1

Make the rocket thrust stable to 350 newtons

```
<?php
$total = 0;
$booster1 = 150;
$booster2 = 200;

function rocketThrust()
{
global $total, $booster1, $booster2;
$total = $booster1 + $booster2;
}

rocketThrust();
echo 'Rocket thrust is currently stable at '.$total.' newtons';
?>
```

Example 1

Week 7 Exercise - Part 2

Add a third rocket booster, and again set thrust stable at 350 newtons

```
<?php
$total = 0;
$booster1 = 50;
$booster2 = 100;
$booster3 = 200;

function rocketThrust()
{
global $total, $booster1, $booster2, $booster3;
        $total = $booster1 + $booster2 + $booster3;
}

rocketThrust();
echo 'Rocket thrust is currently stable at '.$total.' newtons';
?>
```

Example 2

Manipulating Data - intro

- numbers, text, input
- mathematical operations such as addition, subtraction, multiplication...
- string operations such as replace, concatenate, re-order, diff...
- operators such as +, -, and *
- functions are commands that perform more sophisticated calculations such as calculating the square root of a number

Manipulating Data - assignment operator (=)

- nothing more than an equal, =, sign

```
$variableName = value;
```

- value could be a fixed number, specific string, mathematical equation that calculates a single value...

Manipulating Data - using arithmetic operators

+	addition	10 + 22.6
-	subtraction	22.6 - 10
*	multiplication	10 * 4
/	division	120 / 4
%	modulus	39 % 7
-	negation	- 2

Manipulating Data - operator precedence

- normally plays a role in multiple calculations on the same value or variable
- how do we know the order a computer will complete calculations?
- group calculations together to ensure sequence of overall calculation
- bracket calculations to help isolate calculations

100, 75, 25

Manipulating Data - Mathematics

A few mathematics functions

- with basic mathematical operators you can create any type of complicated formula (quadratic equation, generating random numbers...)
- most programming languages include built-in mathematical functions
- functions are either built-in or available as separate libraries
- many advantages to the use of built-in mathematical functions
- mathematical functions remove the need to know how to write a function to calculate the square root, for example

Manipulating Data - Mathematics

A few PHP Mathematics functions - rocket thrust exercise!

- abs() finds the absolute value of a number

- round() rounds a number to the nearest integer

- ceil() rounds a number upwards to the nearest integer

- floor() rounds a number downwards to the nearest integer

- min() returns the smaller of the two arguments passed

- max () returns the larger of the two arguments passed

- rand(min, max) simply returns a random integer
    - Example

```
<?php
echo '<p>'.rand().'</p>';
echo '<p>'.rand(1, 49).'</p>';
?>
```

Further information: PHP Mathematics functions

Manipulating Data - Mathematics

```
<?php
$total = 0;
$booster1 = 50;
$booster2 = 100;
$booster3 = rand();

echo '<p>Original Total = '.$total.'</p>';

function rocketThrust()
{
global $total, $booster1, $booster2, $booster3;
	$total = $booster1 + $booster2 + $booster3;
}

rocketThrust();
echo '<p>Rocket thrust is currently stable at '.$total.' newtons</p>';
?>
```

- rocketThrust Random Example

Manipulating Data - Working with Strings

- many programming languages include built-in string functions such as
    - counting the number of characters in a string
    - removing characters from a string
    - comparing strings
    - splitting strings...

Common functions can include options such as

length(x) = counts the number of characters in a string (x) including spaces
trim(x,y) = removes characters from a string
index(x,y) = returns the positon of a string within another string
compare(x,y) = compares two strings to see where they differ
replace(x,y,z) = replace one string with another string within a main string

Manipulating Data - Working with Strings

PHP - some common functions

- strlen() finds the character length of a string
    strlen("DIGH 401");

- strpos() searches for a given character, word, phrase etc within a string
    strpos("Hello World, again!", "World");

- str_replace() replaces specified characters in a string with new characters, case-sensitive
    - str_replace('World', 'Jim', 'Hello World, again!');

Example 1

Example 2

Manipulating Data - Working with Strings

Regular expressions

- before you can manipulate a string you need to find it, eg: a substring
- some languages include string searching functions
- these functions tend to be fairly limited and normally require exact matches
- regular expression is a series of symbols that tells the computer how to find a given pattern in a string
- regular expressions offer a more powerful way of searching strings

Manipulating Data - Working with Strings

Regular expressions - Single character (.) wildcard

- simplest way to search for a pattern is by using a single character
- single character wildcard (.)

b.t

- use the number of wildcards (.) to match the number of characters searched

b..t would return boat, boot, bolt...

- this expression will only match the exact number of wildcards

DIGH 401 - Introduction to Computing

Manipulating Data - Working with Strings

Regular expressions - Specific character matching

- search for any specific character using [ ] symbols
- enclose the characters to search for within the brackets
- add multiple characters in the brackets to search for different alternatives

```
b[aou]t
```

- this expression will search for words that contain either a, o, or u between b and t
- you can also use the not ^ character to specify characters to ignore

```
b[^ao]t
```

- ^ can also be used in regular expression to define an anchor normally at the start of a target string

```
q[^x]
```

DIGH 401 - Introduction to Computing

Manipulating Data - Working with Strings

Regular expressions - Pattern matching with ranges

- match characters within a range of specified characters
- search for just letters, eg: bu[a-z]
- restrict range as much as is required, eg: bu[r-u]
- you can also check numeric ranges, eg 15[0-9]
- numbers can also be restricted to a shorter range, eg: 15[2-5]
- combine ranges to check wider selection of characters and values

[0-9a-fA-F]

[0-9a-fxA-FX]

Manipulating Data - Working with Strings

Regular expressions - Non-Printable characters

- use special characters to add non-printable characters to an expression

\t = tab
\r = carriage return
\n = line feed or new line

\xFF eg: \xA9 uses the hexadecimal index to match the copyright symbol in the Latin-1 character set

\uFFFF eg: \u20AC uses Unicode to match the Euro currency symbol

- all non-printable characters can be used in regular expressions

Manipulating Data - Working with Strings

Regular expressions - Anchors

- do not directly match any characters
- they match a position in the given string

^	matches at the start of the string
$	matches at the end of the string
\b	matches at a word boundary

\w word characters

\W
\B

Manipulating Data - Working with Strings

Regular expressions - Alternation

- character used is a pipe |
- similar to OR operator in programming languages

```
book | volume
```

- finds either everything to left or everything to the right

```
(book) | (volume)
```

DIGH 401 - Introduction to Computing

Manipulating Data - Working with Strings

Regular expressions - Repetition

- three basic ways to enable repetition in regular expressions
    - ?, * and + operators

- ? makes the preceding token in the regular expression optional

```
colou?r
```

- * attempts to match the preceding token zero or more times

```
<[A-Za-z] [A-Za-z0-9]*>
```

- + attempts to match the preceding token once or more

```
a+b
```

Manipulating Data - Working with Strings

Regular expressions - Combining expressions

- combine multiple expressions to search for a variety of different string patterns

```
b[aei].[0-9]
```

- this example will search through a string looking for matches based on each of the four character requirements

Try www.regular-expressions.info for further info, tutorials, examples etc...

Javascript examples available at the W3 Schools