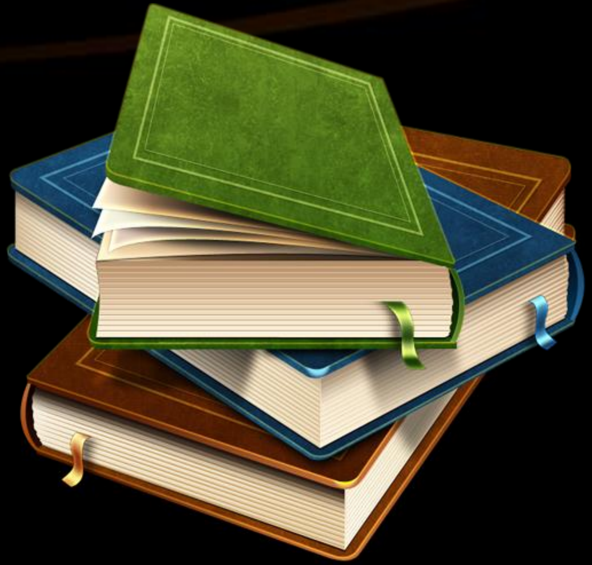
Strings in PHP

Working with Text in PHP
Strings and String Functions

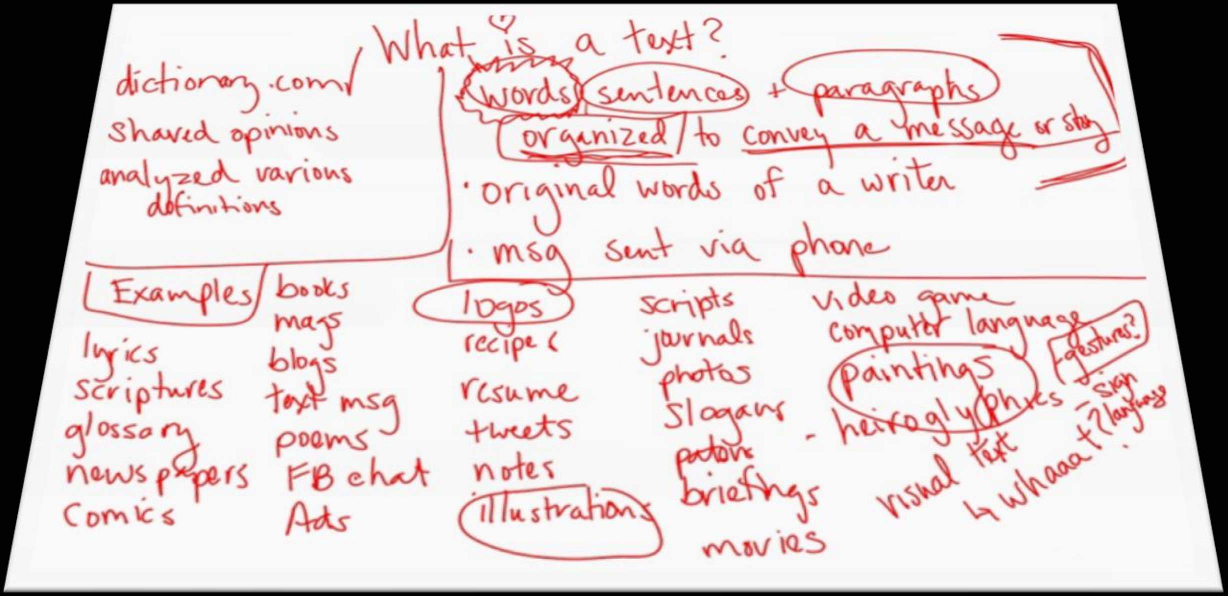
SOFTWARE UNIVERSITY FOUNDATION

Mario Peshev
Technical Trainer
http://peshev.net
Software University
http://softuni.bg

Table of Contents



What Are Strings?



dictionary.com ✓
shared opinions
analyzed various definitions

What is a text?

- words sentences + paragraphs organized / to convey a message or story
- original words of a writer
- msg sent via phone

Examples

lyrics
scriptures
glossary
newspapers
Comics

books
mags
blogs
text msg
poems
FB chat
Ads

logos
recipes
resume
tweets
notes
illustrations

scripts
journals
photos
Slogans
patterns
briefings
movies

video game
computer language
paintings
gestures?
heiroglyphics
sign language
visual text
↳ whaaat?

What Are Strings?

- Text string:
  - Contains zero or more characters surrounded by double or single quotation marks
  - Can be used as literal values or assigned to a variable

```
<?php
echo '<p>Mr. Svetlin Nakov</p>';
$workPlace = "<span>Software University</span>";
echo $workPlace;
?>
```

  - Can also be surrounded with single quotation marks

What Are Strings? {2}

- Single quotes are escaped when used in double quoted strings

```php
<?php
echo "<p>I'm a Software Developer</p>";
?>
```

- Double quotes are escaped when used in single quoted strings

```php
<?php
echo '<span>At "Software University"</span>';
?>
```

Manipulating Strings

String Operators

- In PHP, you use two operators to combine Strings
  - Concatenation Operator "."
  - Concatenation assignment operator ".="

```
<?php
$homeTown = "Madan";
$currentTown = "Sofia";
$homeTownDescription = "My home town is" . $homeTown;
$homeTownDescription .= "But now I am in " . $currentTown;
echo $homeTownDescription;
?>
```

Escape Characters

- Added before a special purpose character follows it has a special purpose
- In PHP, the escape character is the backslash \

```
<?php
$myCourse = 'I\'m a PHP Developer';
?>
```

Escape Sequence

- The escape character combined with one or more other characters is called an escape sequence

Escape Sequence	Description
\\	Insert a backslash
\$	Insert a dollar sign
\r	Insert a carriage return
\”	Escape a double quotation mark
\t	Insert a horizontal tab
\n	Insert a new line

Simple and Complex String

- Simple string syntax uses the value of a variable within a string by including the variable name inside a text string with double quotation marks
- When variables are placed within curly braces inside of a string, it is called complex string syntax

```php
<?php
$popularName = "Pesho";
echo "Hello $popularName";
?>
```

```php
<?php
$popularName = "Pesho";
echo "Hello {$popularName}";
?>
```

Simple and Complex String{2}

- When variables are placed within curly braces inside of a string, it is called complex string syntax

```php
<?php
$popularName = "Pesho";
echo "Hello {$popularName}";
?>
```

Manipulating Strings

Live Demo

Built-in String Functions

Counting Characters

- The most commonly used string counting function is the `strlen()` function
- returns the total number of characters in a string

```php
<?php
$name = "Software University";
echo strlen($name);
?>
```

Output : 11

Counting Words

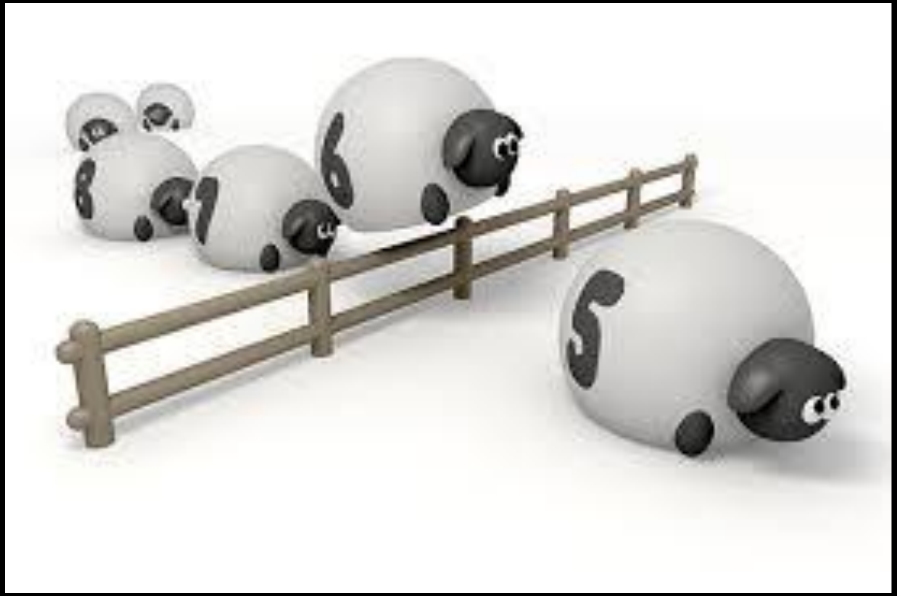
- The `str_word_count()` function returns the number of words in a string
- Pass the `str_word_count()` function a literal string or the name of a string variable whose words you want to count

```php
<?php
$countries = "Bulgaria, Brazil, Italy, USA, Germany";
echo "<p>String contains " . str_word_count($countries). " countries.</p>";
?>
```

String contains 5 countries.

Counting Strings

Live Demo



Finding Characters and Substrings

- There are two types of string search and extraction functions:
  - Functions that return a numeric position in a text string
    - `strpos()` - Performs a case-sensitive search and returns the position of the first occurrence of one string in another string

```php
<?php
$countries = "Brazil, Italy, Bulgaria, USA, Germany";
$bulgaria = "Bulgaria";
echo "Position of Bulgaria is: " . strpos($countries, $bulgaria);
?>
```

Position of Bulgaria is: 15

Extracting Characters and Substrings

- Functions that return a character or substring
  - `strstr()` - Function starts searching at the beginning of a string
  - `strstr($input, $search, true)` - Function starts searching at the end of a string
- Both functions return a substring from the specified characters to the end of the string

```
<?php
$countries = "Brazil, Italy, Bulgaria, USA
$bulgaria = "Bulgaria";
echo "String from Bulgaria to end: " . strstr($countries, $bulgaria);
?>
```

String from Bulgaria to End: Bulgaria, USA, Germany

Extracting Characters and Substrings{2}

- `substr()` - To extract characters from the beginning or middle of a string
- You pass to the `substr()` function a text string along with the starting and ending positions of the substring you want to extract

```
<?php
$email = "nakov@example.com";
$nameEnd = strpos($email, "@");
echo "The name portion of the e-mail address is: " . substr($email, 0,$nameEnd);
?>
```

```
The name portion of the e-mail address is: nakov
```

Finding And Extracting

Live Demo

String Replacing

- The `str_replace()` and `str_ireplace()` functions both accept three arguments:
  - The string you want to search for
  - A replacement string
  - The string in which you want to replace characters

```php
<?php
$email = "bignakov@example.com";
$newEmail = str_replace("bignakov", "juniornakov", $email);
echo $newEmail;
?>
```

juniornakov@example.com

Trim

- Trim - Strip whitespace (or other characters) from the beginning and end of a string

```
<?php
$boo = "   foo   ";
echo "After trim "  trim($boo);
?>
```

After trim “foo”

- This function returns a string with whitespac ... g and end of $boo.
- Also have:
  - ltrim() – trim from beginning of a string
  - rtrin() – trim from end of a string

Case Changing

- strtolower() - Make a string lowercase

```php
<?php
$boo = "FOO";
echo strtolower($boo);
?>
```

foo

- strtoupper() - Make a string uppercase

```php
<?php
$boo = "foo";
echo strtoupper($boo);
?>
```

FOO

Converting String to Array

- The `str_split()` function splits each character in a string into an array element
- The length argument represents the number of characters you want assigned to each array element

```
$array = str_split(string[, length]);
```

- The `length` argument represents the number of characters you want assigned to each array element

Converting String to Array{2}

- The explode() function splits a string into an indexed array at a specified separator

```
<?php
    $presidents = "Georgi Pyrvanov;Jelio Jelev;Petyr Stoqnov;Rosen Pleveneliev;";
    $presidentAsArray = explode( ";“ , $
     print_r( $presidentAsArray);
?>
```

```
Array
(
    [0] => Georgi Pyrvanov
    [1] => Jelio Jelev
    [2] => Petyr Stoqnov
    [3] => Rosen Pleveneliev
)
```

explode() Function

- Does not separate a string at each character that is included in the separator argument
- Evaluates the characters in the separator argument as a substring
- If you pass to the `explode()` function an empty string as the separator argument, the function returns a value of false

Converting Array to String

- Implode() - Combines an array’s elements into a single string, separated by specified characters

```php
<?php
$presidents = ["Georgi Pyrvanov", "Jelio Jelev", "Petyr
Stoqnov", "Rosen Pleveneliev"];
$presidentAsString = implode(";", $presidents);
echo $presidentAsString;
?>
```

```
Georgi Pyrvanov;Jelio Jelev;Petyr Stoqnov;Rosen Pleveneliev
```

String <> Array

Live Demo

String Comparison Functions

- The `strcasecmp()` function performs a case-insensitive comparison of strings
- The `strcmp()` function performs a case-sensitive comparison of strings
- Both functions accept two arguments representing the strings you want to compare
- Most string comparison functions compare strings based on their ASCII values

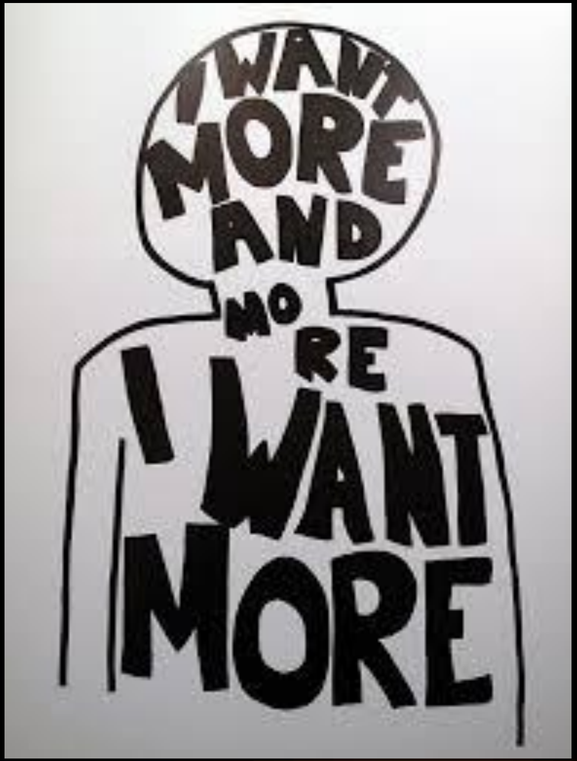
String Comparison Functions Example

```php
<?php
    $fName = "Nakov";
    $fNameSmall = "nakov";
    echo "Case insensitive\n";
    echo strcasecmp($fName, $fNameSmall) . "\n";
    echo "Case sensitive\n";
    echo strcmp($fName, $fNameSmall);
?>
```

```
Case insensitive
0
Case sensitive
-1
```

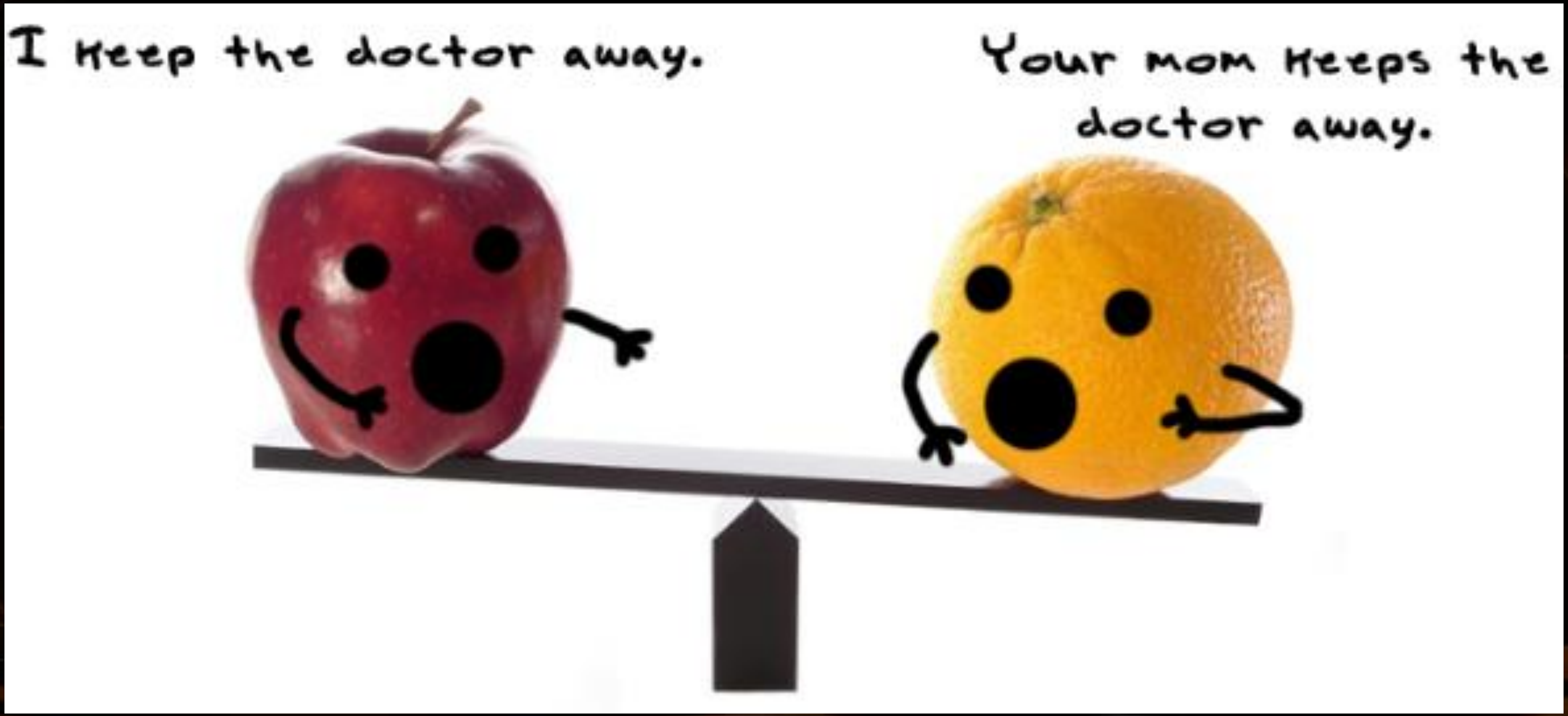
More comparing functions

- `strnatcmp()`, `strncasecmp()`, `levenshtein()`, `metaphone()`, `similar_text()`, `soundex()`, `strnatcasecmp()`

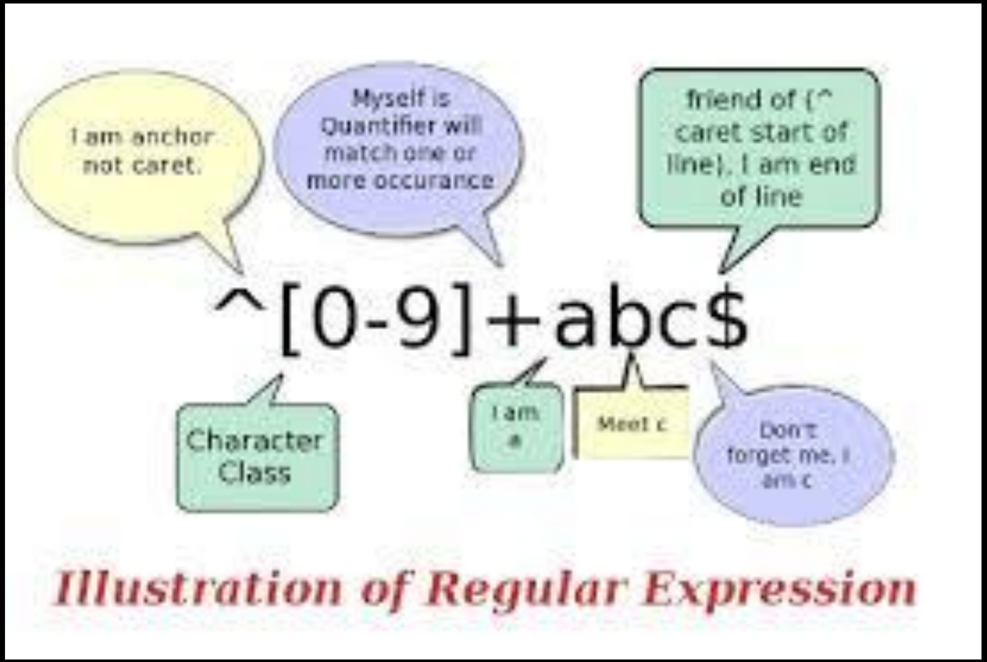


String Comparing

Live Demo



Regular Expressions


I am anchor not caret.
Myself is Quantifier will match one or more occurance
friend of (^ caret start of line), I am end of line
^[0-9]+abc$
Character Class
I am a
Meet c
Don't forget me, I am c
Illustration of Regular Expression

Regular Expressions

- It is usually possible to use a combination of various built-in PHP functions to achieve what you want.
- However, sometimes this gets complicated and we turn to Regular Expressions.
- Regular expressions are a concise (but complicated!) way of pattern matching
- Define a pattern used to validate or extract data from a string

Some definitions

- Definition of the pattern (the ‘Regular Expression’):
  - '/^[a-z\d\._-]+@([a-z\d-]+\.)+[a-z]{2,6}$/i‘
- PHP functions to do something with data and regular expression:
  - `preg_match()`, `preg_replace()`

Regex: Delimiters

- The regex definition is always bracketed by delimiters, usually a ‘/’:

```
pattern: ’/php/’;

Matches: ‘php’, ‘I love php’, ‘phpphp’

Doesn’t match: ‘PHP’, ‘I love ph’
```

Regex: Character groups

- A regex is matched character-by-character. You can specify multiple options for a character using square brackets:

```
$regex = ’/p[huo]p/’;

Matches: ‘php’, ’pup’, ‘pop’

Doesn’t match: ‘phup’, ‘ppp’, ‘pHp’
```

Regex: Predefined Classes

- A regex is matched character-by-character. You can specify multiple options for a character using square brackets:

\d	Matches a single character that is a digit (0-9)
\s	Matches any whitespaces character (include tabs and line breaks)
\w	Matches any alphanumeric character (A-Z, 0-9) or underscore

Regex: the Dot

- The special dot character matches any character except for a line break:

```
$regex = ’/p.p/’;

Matches: ‘php’, ’p&p’, ‘p(p’, ‘p3p’, ‘p$p’

Doesn’t match: ‘PHP’, ‘phhp’
```

Regex: Repetition

- There are a number of special characters that indicate the character group may be repeated:

?	Zero or 1 times
*	Zero or more times
+	1 or more times
{a, b}	Between a and b times

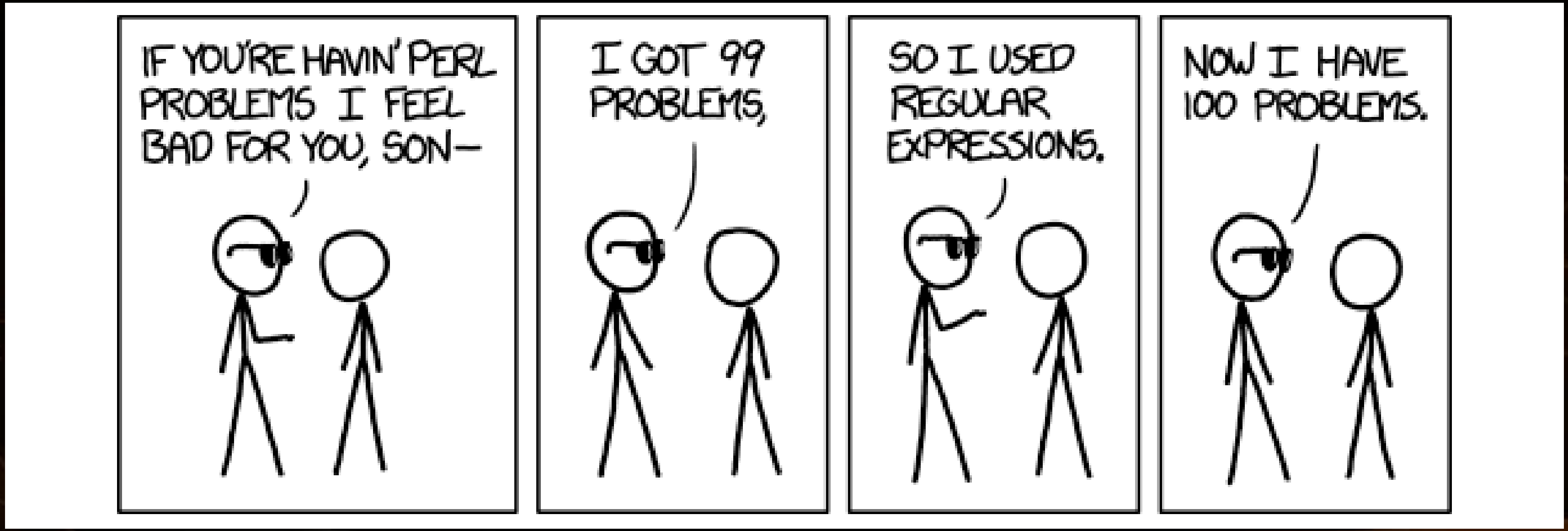
Regex: Anchors

- So far, we have matched anywhere within a string. We can change this behavior by using anchors:

^	Start of the string
$	End of string

Regular Expressions

Live Demo

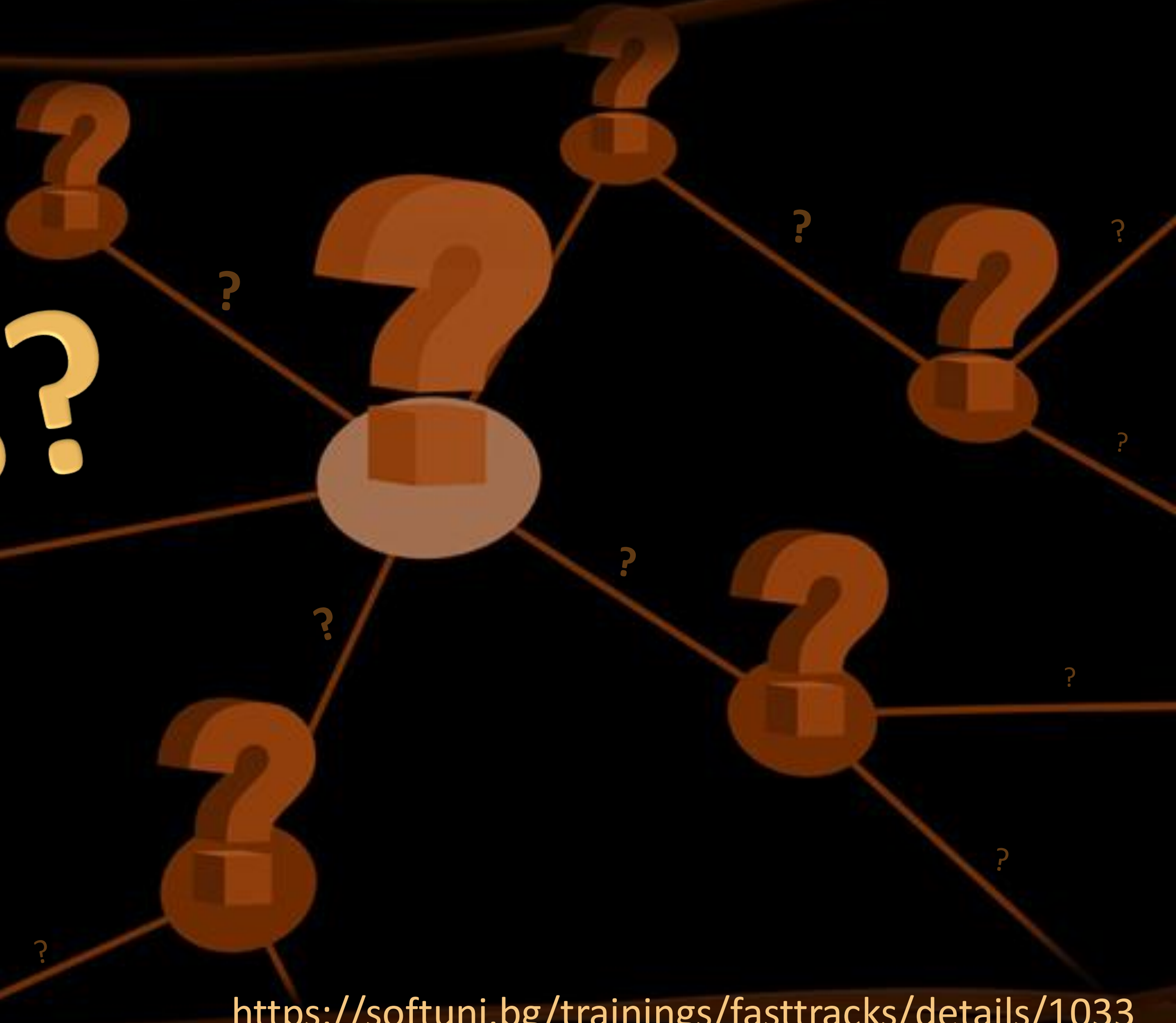

IF YOU'RE HAVIN' PERL PROBLEMS I FEEL BAD FOR YOU, SON—
I GOT 99 PROBLEMS,
SO I USED REGULAR EXPRESSIONS.
NOW I HAVE 100 PROBLEMS.

Summary

- All about simple strings
- Manipulating Strings
  - Escaping, Operators
- Built-in String Functions
  - Most popular functions in PHP
- Regular Expressions
  - Regex Pattern
  - preg_match(), preg_replace()

PHP & MySQL

SOFTWARE UNIVERSITY FOUNDATION



Questions?

https://softuni.bg/trainings/fasttracks/details/1033

License

Free Trainings @ Software University

- Software University Foundation – softuni.org
- Software University – High-Quality Education, Profession and Job for Software Developers
  - softuni.bg
- Software University @ Facebook
  - facebook.com/SoftwareUniversity
- Software University @ YouTube
  - youtube.com/SoftwareUniversity
- Software University Forums – forum.softuni.bg