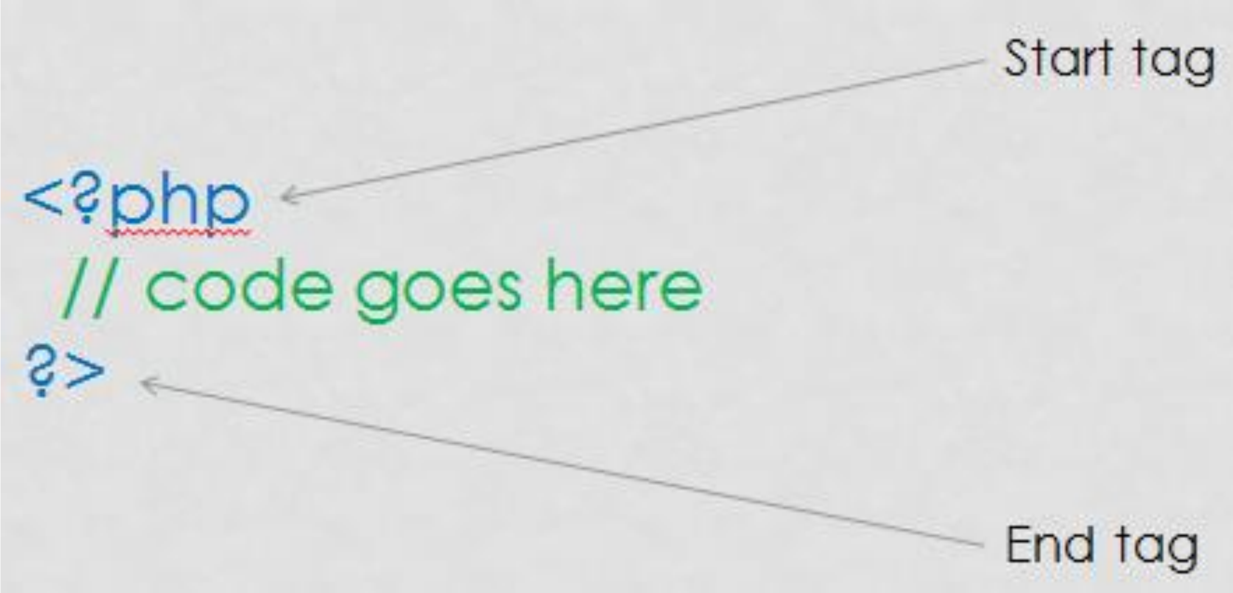
PHP SYNTAX

MET CS602
Andrew Sheehan

PHP
TAGS



```
<?php
  // code goes here
?>
```

Start tag

End tag

You can use as many as needed.

EMBEDDING
IN HTML

```
<header class=“top_menubar”>
  <h1 class=“r2 stg_t12a”>Welcome
      <?php echo($_SESSION[‘username’]);  ?>
  </h1>
</header>
<section id=“root”>
  …
</section>
```

ENDING YOUR STATEMENT TERMINATION

```
<?php
    $message =
        “The force is strong with this one.”;
 ?>
```

PHP
CASE SENSITIVITY ON LANGUAGE KEYWORDS

```
<?php
   while ( true ) { … }
                                  // okay
   WHILE ( true ) { … }
?>
```

VARIABLES

DECLARING

```
<?php
  /*  Not strongly-typed, like C++ or Java. */
   $message =
            “The force is strong with this one.”;
?>
```

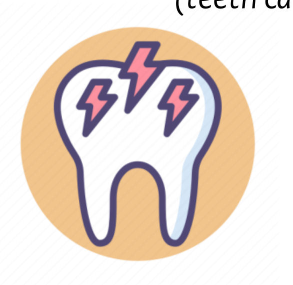
VARIABLES
ARE SENSITIVE

```
<?php
 // variables are case SenSitive

$message =
        “The force is strong with this one.”;

 $Message =
        “This is not the same as the first one…”;
 ?>
```

(teeth can be very sensitive…)



VARIABLES/DATA
8 TYPES

String

boolean

Integer

NULL

Array

Object

Double

Resource

PHP
COMMENTS

```
<?php
  // single line comment
  $a = 1;

/* multi-line
    comment */
   $results = fetch(..);
?>
```

VARIABLES
3 SCOPES

local	Declared in a function
global	Declared outside of a function
static	Use it everywhere

THE NULL
RULES ON NULL

When you create a variable without assigning it, the value is null.

- Returns FALSE when tested with isset().
- Is case insensitive.
- Evaluates to FALSE in boolean context.

VARIABLES
ABOUT RESOURCE TYPES

A variable you have when opening a database or file.

Means: External resource. Ready for operations

VARIABLES
ABOUT STRINGS

Single quote use: interpreted literally.

```
<?php
 /*  Variable interpolation with double quoted strings. */
  $who = “this one.”;

  $message =
          “The force is strong with $who”;
?>
```

PHP
COMMON ESCAPE SEQUENCES

\$: dollar sign

\n: new line

\t: puts a tab (5 spaces?)

\\: single backslash character

\”: single double quote mark

PHP CONSTANTS

define (name, value)

Names of constants follow standard PHP naming rules.

$ is not needed for constants.

PHP
DISPLAYING OUTPUT

echo (value) // many strings

print (value) // 1 string

Only 2 ways to response to clients

PHP
OPERATORS

Yes, there are operators.
You know them already.