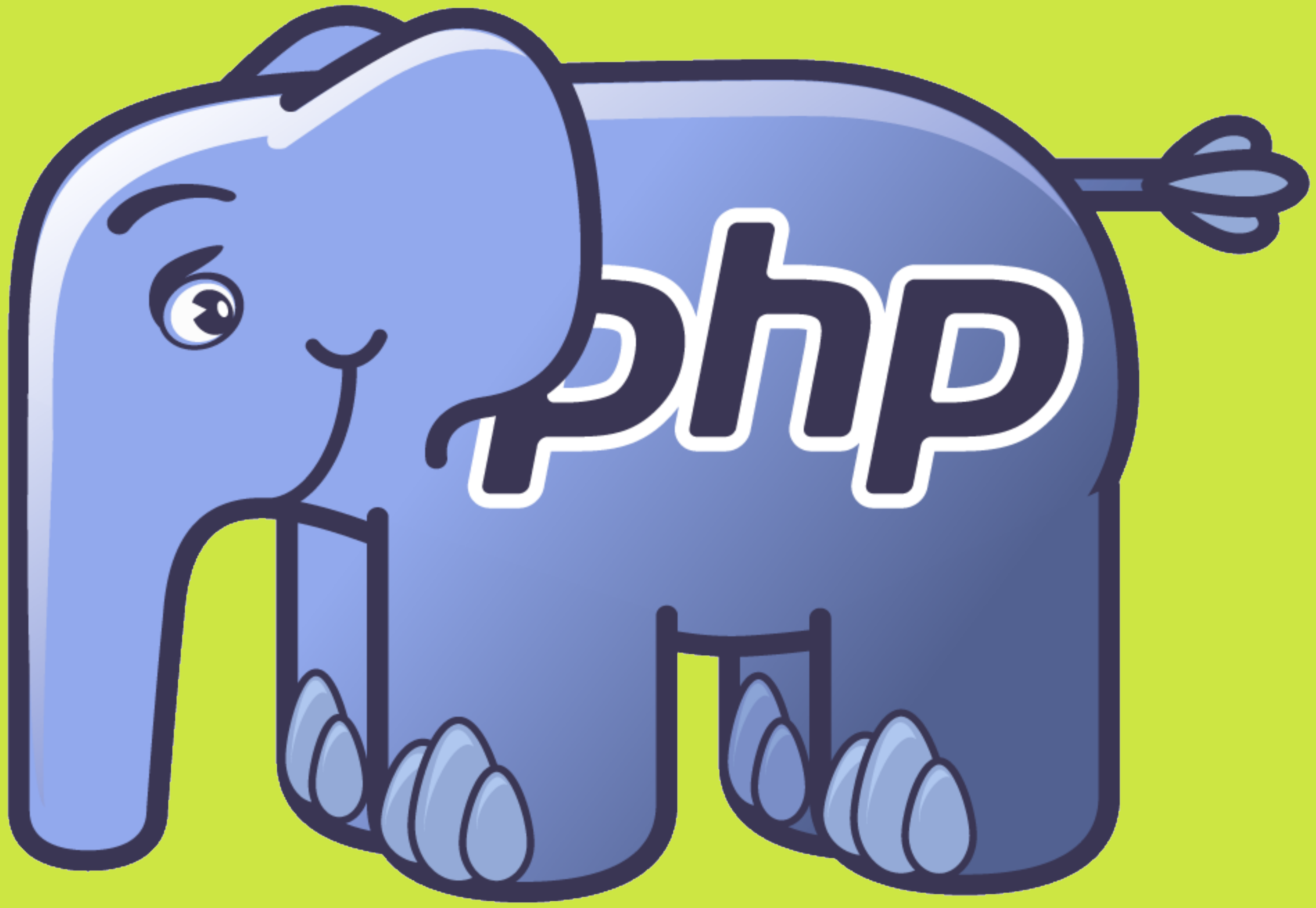
Fundamentals

VERSION .03


php

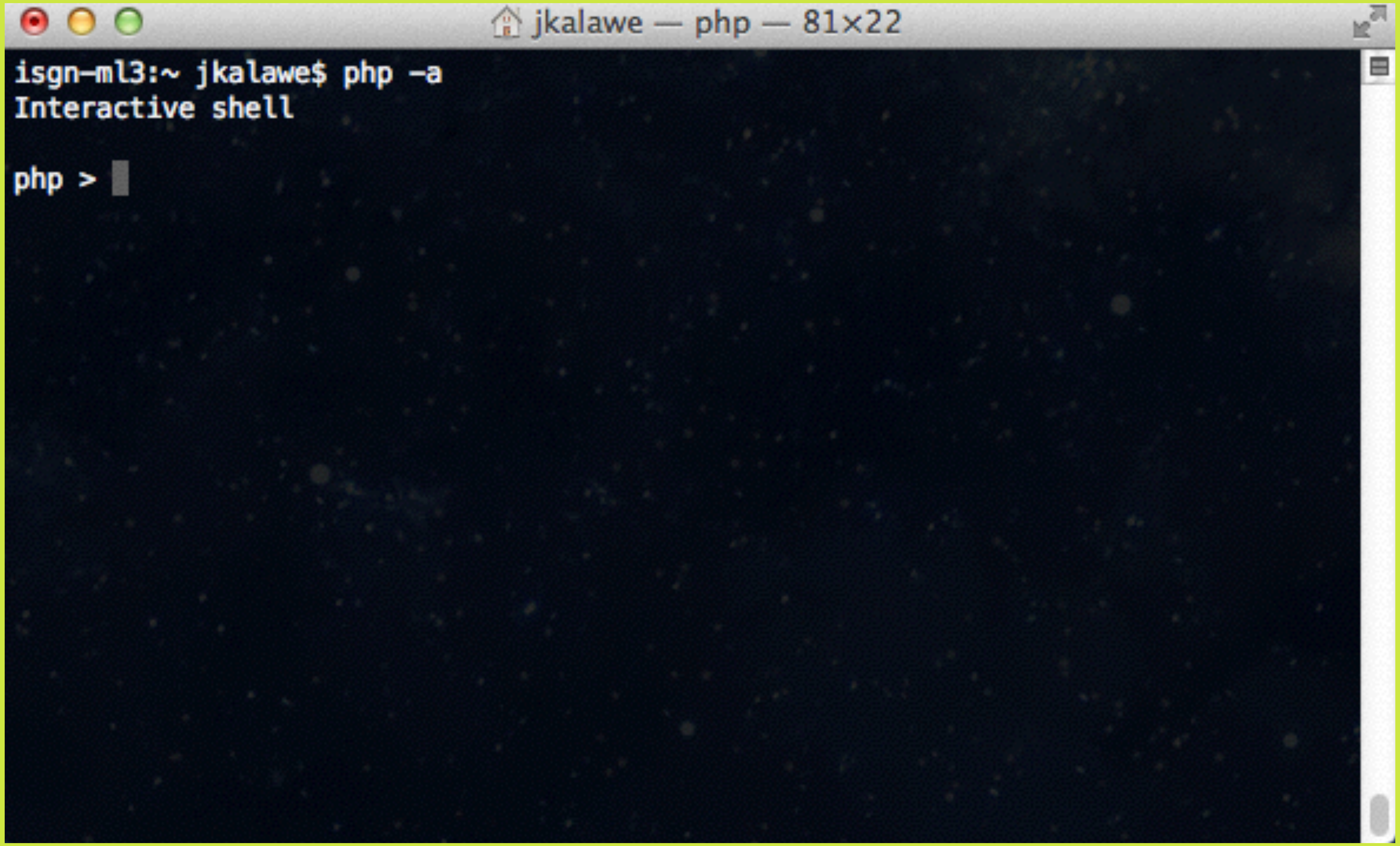
TERMINOLOGY

PHP Interactive Shell

Type and execute code in the shell

Open your shell window and type the following

```
php -a
```



```
isgn-ml3:~ jkalawe$ php -a
Interactive shell

php > 
```

TERMINOLOGY

Every line must end with a semicolon

Let's output a text string

```
echo “hello world”;
```

Primitive Data Types

TERMINOLOGY

Primitive Data Type

Building blocks for any language

Open your shell window and type the following

```
php -a
```

TERMINOLOGY

String

Series of characters.

Boolean

True of false.

TERMINOLOGY

Integer

Whole number: 1,2,3,4

Float

Real number: 1.1, 2.3, -1.1

Variables

TERMINOLOGY

Variable

Storage & reference of a value.

Must start with a dollar sign

```
$test
```

FOLLOW ALONG

Assign an integer to a variable

```
$int = 1;
```

Output the value of the variable

```
echo $int;
```

FOLLOW ALONG

Assign a string to a variable

```
$string = “I don’t like this cheese”;
```

Output the value of the variable

```
echo $string;
```

FOLLOW ALONG

Assign a boolean to a variable

```
$bool = TRUE;
```

Output the value of the variable

```
echo $bool;
```

Operators

TERMINOLOGY

OPERATOR

Something that takes a value and outputs another value

FOLLOW ALONG

String concatenation

```
$concat = $string . $int . $bool;
```

```
echo $concat;
```

String concatenation assignment

```
$string = $string .= “next line”;
```

```
echo $string;
```

FOLLOW ALONG

Arithmetic

```
echo 2+2;
```

```
echo 1*2;
```

Logic

```
echo 100 > 50;
```

```
echo 100>=50;
```

conditionals

TERMINOLOGY

if

Evaluates if expression is true

Let's see if 100 is greater than 50

```
if( 100 > 50) {
  echo ‘100 is greater than 50’;
}
```

TERMINOLOGY

else

Executes when if statement fails.

FOLLOW ALONG

Let's return a message if the expression fails

```
if( 100 < 50) {
  echo ‘100 is greater than 50’;
} else {
  echo ‘100 is less than 50’;
}
```