

PHP

`http://localhost/DIR-NAME/abc.php`

Comment

Singeline: //

Multiline: /* */

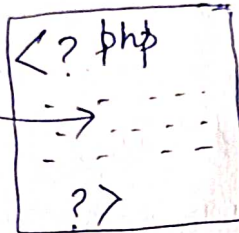
Run: ↗

```
/* echo "I am at CST department"
echo " . . . . . "
echo " . . . . . " */
```

Variable

→ (\$)

```
$a = 10;
echo $a
```



PHP variables are
case sensitive
i.e. \$abc \$ABC
are different

```
$a = 10;
```

```
$b = 20;
```

i. echo "\$a + \$b" (sum)

ii. echo "\$a . \$b" (concatenation)

iii. echo "\$a . " . \$b" (Adding space)

Adding white spaces:-

iv. \$space = str_repeat(' ', 10) → creating a string of 10 white spaces
echo \$a . \$space . \$b

Adding newline:-

v. echo "\$a . "
" . \$b

or,

```
echo $a;
echo "<br>";
echo $b;
```

PHP Datatypes

integer

```
$x = 10;  
echo $x;
```

```
var_dump($x);
```

int(10)

Returns datatype &
value of a variable

string

```
$x = "Hello world"
```

```
var_dump($x) ——— string(11)
```

datatype

No. of characters

float

```
$x = 10.2; —————> float(10.2)
```

```
var_dump($x);
```

boolean

```
$x = true
```

```
$y = false
```

boolean(true)

NULL value

```
$x = NULL
```


Object

Class Complex

```
{
    private $x;
    private $y;
    public function __construct()
    {
        $this->x = 10;
        $this->y = 20;
    }
    public function display()
    {
        return $this->x . " + i " . $this->y;
    }
}
```

Constructor

Complex

x
y
display()

```
$c = new Complex();
echo $c->display();
```

var_dump(&c)

Information about the class

Array

```
$x = array(10, 12, 20, 91, 57, 68);
```

```
print_r($x);
```

```
var_dump($x);
```

```
echo $x[0]; ✓
```

```
echo $x[1]; ✓
```

✓ using loop

```
foreach ($x as $value)
```

```
    echo " $value ";
```

```
for($i=0; $i<6; $i++)
```

```
{ echo $x[$i]. " ";  
}
```

For loop

PHP Strings

```
echo strlen("Hello World"); _____ 11
```

```
echo str_word_count("Hello World"); _____ 2
```

```
echo strtolower("Hello World"); _____ Hello world
```

```
strpos("Hello World", "World"); _____ 6
```

Character
position of
first match

H	e	l	l	o
0	1	2	3	4

- position starts from 0
- case sensitive

```
$x = echo str_replace("World", "India", "Hello World");
```

```
echo $x;                      O/P → Hello India
```


Numbers

```
$x = 90;
```

```
var_dump($x); _____ int(90)
```

```
var_dump(is_int($x)); _____ bool(true)
```

is_int()

is_float()

is_long()

PHP infinity

PHP_FLOAT_MAX
largest floating point number in PHP

If any number is larger than PHP_FLOAT_MAX
then it is considered as infinity

1.79e308

```
$x = 1.79e400;
```

```
var_dump(is_finite($x));
```

$a \times 10^b$

→ bool(false)

```
$x = 1.79e40; var_dump(is_finite($x));
```

→ bool(true)

```
$x = 91;  
var_dump(is_numeric($x)); _____ bool(true)
```

```
$x = "Hello"  
var_dump(is_numeric($x)); _____ bool(false)
```

Casting string to integer

```
$x = "23"
```

```
$y = (int) $x;
```

Float → Integer

```
$x = 23.7;
```

```
$y = (int) 23.7;
```

PHP Math

```
$x = rand();  
echo $x;
```

```
rand();  
rand(10, 100);  
pi();  
min(x1, x2, x3, ...);  
max(x1, x2, x3, ...);  
abs();  
sqrt();  
round();
```

```
sin(), cos(), tan();  
pow();
```

```
log(), log10();  
exp(x) → ex
```

```
ceil(); floor();
```

```
base_convert(); → base_convert("E196", 16, 8);
```

```
deg2rad(); → deg2rad("45");
```

~~PHP Constants~~

~~define(constant name, value, case sensitivity);~~

~~define("A", "Apple", true);~~

~~echo A;~~

~~echo a;~~

PHP Constants

Constants are global

define (constant name, value);

```
define ("A", "Apple");
```

```
echo A;
```

Constant array

```
define ("A", ["Apple", "Mango", "Banana"]);
```

```
echo A[0];
```

```
echo A[1];
```

```
echo A[2];
```

?? operator

$\$x = \$a ?? \$b;$

if $\$a$ exists then $\$x = \a
otherwise $\$x = \b

? : operator
(Ternary)

~~$\$x = \$a ? \$b :$~~ $\$x = \text{expression} ? \$a : \$b;$

- Comparison
- Arithmetic
- Logical
- Assignment | Preincrement | Postincrement
- Array
- String Concatenation

Array operators

```
$a = array(100, 200, 300);
```

```
$b = array(10, 90);
```

```
$c = $a + $b
```

```
✓ print_r($c);
```

```
✓ echo $c[0];
```

+ UNION

100	200	300
10	90	

100	200	300
10	90	
<hr/>		
100	200	300

 ✓

```
$a = array(100, 200, 300);
```

```
$b = array(11, 17, 23, 99, 68, 70);
```

```
$c = $a + $b;
```

100	200	300
-----	-----	-----

11	17	23	99	68	70
----	----	----	----	----	----

<hr/>					
U:	100	200	300	99	68 70

 ✓

```
$a = array(100, 200);
```

```
$b = array(0, 0, 11, 17, 23);
```

```
$c = $a + $b;
```

100	200
-----	-----

0	0	11	17	23
---	---	----	----	----

<hr/>				
U:	100	200	11	17 23

== Equality

```
var_dump($a == $b);
```

→ bool(true)

bool(false)

Identity ===

```
$a = array(10, 20);  
$b = array("10", "20");
```

```
var_dump($a === $b);  
var_dump($a == $b);
```

Non-identity

!==

Not equal to

!=

■ If - else

■ If - elseif - else

■ switch

■ Loop — while, do-while, for, foreach

foreach works only on arrays

```
foreach($colour as $value)  
    echo $value;
```

break, continue

■ Function

argument | default argument | ..

```
function functionName( )  
{  
    Body  
}
```

■ PHP Superglobals

Predefined
variables

```
$GLOBALS, $_SERVER  
$_GET, $_POST, $_SESSION[]  
$_COOKIE[]
```

File inclusion

```
include ("myModule.php");  
f(10,20); <
```

In myModule.php

```
<?  
function f($a,$b)  
{ return $a+$b;  
}  
?>
```

PHP Session

A group of interactions between client and website within a given time frame.

A series of requests ^{that} comes from ~~client~~ the same client during certain time period.
Session IDs are randomly generated in PHP.

- session_start(); Starting a new session.
- session_destroy(); Destroying session data | Destroying a complete session
- Storing data for the lifetime of a session using `$_SESSION[]` superglobal array
eg. `$_SESSION["Name"] = "Vijay";`
`$_SESSION["Roll"] = 11;`
- `isset($_SESSION["Name"]);` — checking if "Name" is set
- `unset($_SESSION["Name"]);` — destroying a session data "Name"

PHP Cookies

A small file sent by server to client browser

Browser stores this information to client machine for future use.

When next time the browser sends request to the server, the cookies information is also sent.

The server uses the information to identify user.

```
setcookie();  
setcookie("name", "Vijay", time(), "/", 0);  
  
echo $_COOKIE["name"];  
  
isset($_COOKIE["name"]);
```

Deleting cookie

```
setcookie("name", "", time()-60, "/", "", 0);
```

delete with the date
already expired.