[EXPERIMENT - 5]

Page No.
Date

— IN (PHP) THERE ARE NUMEROUS STRING RELATED FUNCTIONS TO CHANGE AND MANIPULATE STRINGS EFFECTIVELY.

a] strlen()

→ RETURNS THE LENGTH OF A GIVEN STRING

SYNTAX

— int strlen (string $string)

Eg $str = "Hello World";
echo strlen ($str); // 13

b] strtolower()

→ CONVERTS A STRING TO LOWERCASE

SYNTAX

— string strtolower (string $string)

Eg. $str = "Hello Mr. White";
echo strtolower ($str); // hello mr. white

Page No.
Date

c] strtoupper()

→ CONVERTS A STRING TO UPPERCASE

SYNTAX

— string strtoupper (string $string)

Eg. $str = "Hello"
echo strtoupper ($str); // HELLO

d] substr()

→ RETURNS A PART OF A STRING

SYNTAX

— string substr (string $string,
int $start
[int $length])

$string — main string
$start — replace / return from index
$length — OPTIONAL, how long from $start

Eg. $str = "H
echo s
// Hello

e] str_replace

→ REPLACES
STRING

SYNTAX

— str_replace

$search
$replace
$string
$count

Eg. $str =
echo

Page No.
Date

Eg. $str = "Hello World";
echo substr ($str, 0, 5);
// Hello

c] str_replace ()

→ REPLACES ALL OCCURANCES OF SEARCH STRING WITH REPLACE STRING

SYNTAX

str_replace ($search,
$replace,
$string
[$count]
)

$search — VALUE TO SEARCH
$replace — REPLACMENT VALUE
$string — STRING TO OPERATE
$count — HOLDS NUMBER OF MATCHED AND REPLACED VALUES

Eg. $str = "Hello World";
echo str_replace ("Hello", "Hi", $str);

Page No.
Date / /

f] strpos()

→ FINDS POSITION OF FIRST OCCURANCES OF A SUBSTRING IN A STRING.

SYNTAX

— int || false strpos ($haystack ,
$needle ,
[$ int $offset = 0]
)

Eg.

$haystack — STRING TO SEARCH
$ needle — SUB STRING TO FIND
$offset — IF NOT ZERO THEN IT WILL START FROM SPECFIED INDEX. [OPTIONAL]

Eg.

$str = "Hello World";

echo strpos($str, "World");

// 6

g] trim()

→ STRIPS W
END OF

SYNTAX

string trim

$str —
[$character

Eg. $ str

echo

h] explode(

→ SPLITS

SYNTAX

string ex

g) trim()

→ STRIPS WHITESPACE FROM ~~A~~ START TO END OF STRING, RETURNS A STRING

SYNTAX

string trim ($str,
[$character_mask) =
" \t\n\r\0\x0B"
])

$str — STRING TO TRIM
[$character_mask] — SPECIFY STRIPPED CHAR.

Eg. $ str = "___ Hello ___ \n
___ World ___ ";
echo trim ($str);

h) explode()

→ SPLITS A STRING BY A STRING. RETURNS AN ARRAY

SYNTAX

string explode ($string, $str,
[$limit])

Page No.
Date

$string_s — SEPERATOR STRING
$string — INPUT STRING
$limit — OPTIONAL, IF POSITIVE THE RETURNED ARRAY WILL CONTAIN A MAX. OF LIMIT ELEMENTS WITH LAST ELEMENT CONTAINING REST OF STRING. IF NEGATIVE ALL ARE RETURNED EXCEPT LAST.

Eg. $str = "apple, banana, orange";
$arr = explode (", " $str);

i) ucfirst()

→ CONVERTS FIRST CHARACTER TO UPPERCASE.

SYNTAX

ucfirst ($string)

Eg.

$str = "Hello World";

echo ucfirst ($str);

i) ucwords()

→ CONVERTS TO UPPERC

SYNTAX

string ucwo

Eg. $str = "Hell
echo uc

ii) strrev()

→ REVERSES RETURNS

SYNTAX

string str

Eg.

$str = "Hi

echo

Page No.
Date

i) ucwords()

→ CONVERTS FIRST CHAR OF EACH WORD TO UPPERCASE, RETURNS A STRING

SYNTAX

string ucwords($str);

Eg. $str = "Hello world"
echo ucwords($str);

ii) strrev()

→ REVERSES A STRING. RETURNS A STRING.

SYNTAX

string strrev(string $str)

Eg.

$str = "Hi Mom"

echo strrev($str);

Page No.
Date / /

9) str_repeat()

→ REPEATS A STRING SPECIFIED NUMBER OF TIMES.
RETURNS A STRING.

SYNTAX

— string str_repeat (
$input
$multiplier
)

$input — ORIGINAL STRING
$multiplier — TIMES TO REPEAT

```
$str = "abc";
echo str_repeat ($str, 3); // abc abc abc
```

PROCESS	PRODUCT	TOTAL	SIGN
09	15	24	+

[EXPERIMENT 6]

[FUNCTIONS IN PHP]

- FUNCTIONS ARE BLOCKS OF REUSABLE CODE THAT PERFORM A SPECIFIC TASK.

- THEY HELP ORGANIZE CODE, IMPROVE READABILITY, PROMOTE CODE REUSE.

- DELCARARED VIA "function" KEYWORD FOLLOWED BY THE FUNCTION NAME FOLLOWED BY PARATHESIS AND "()" ARGUMENTS IF NEEDED WITH BODY ENCLOSED WITHIN CURLY BRACES "{}"

- FUNCTIONS MAY OR MAY NOT RETURN VALUES. THESE VALUES DEFINE THE RETURN TYPE OF THE FUNCTIONS.

- SCOPE OF VARIABLE IS EFFECTED WITH FUNCTIONS. OUTSIDE VARIABLES ARE GLOBAL AND VARIABLES INSIDE FUNCTIONS ARE LOCAL VARIABLES.

SYN

```
function function_name (arg1, arg2 .... argN) {
    // body
}
```

Eg
```
function greet($n){
    return "Hello $n";
}

echo greet("Spectre");
```

[PASS BY REFRENCE]

- IF WE PASS A PARAMETER BY REFRENCE, YOU ARE PASSING A REFRENCE TO THE ORIGINAL VARIABLE.
- THIS MEANS THAT CHANGES MADE TO THE PARAMETER INSIDE THE FUNCTION AFFECT THE ORIGINAL VARIABLE OUTSIDE THE FUNCTION.

Eg.
```
function increment(&$num){
    $num++;
}

$val = 5;

echo $val; // 5

increment($val);
echo $val; // 6
```

VARIABLE FUNCTION

- IN (PHP), VARIABLE FUNCTIONS ALLOW US TO TREAT A FUNCTION NAME AS IF IT WAS A VARIABLE.

- THE FUNCTION NAME IS RESOLVED AT RUNTIME BASED ON THE VALUE OF THE VARIABLE

- AT CALLING WE CAN PASS ARGUMENTS IF NEEDED OR NOT.

Eg

```
function sayHello() {
    echo "Hello";
}

function sayBye() {
    echo "Bye";
}

$fname = "sayHello";
$fname();          // Hello

$fname = "sayBye";
$fname();          // Bye
```

ANONYMOUS FUNCTIONS

- IN (PHP) ANONYMOUS FUNCTIONS ALLOW US TO CREATE FUNCTIONS WITHOUT A SPECIFIC NAME.

- THEY ARE USEFUL FOR SITUATIONS WHERE YOU NEED TO CREATE A SMALL, ONE-TIME-USE.

- WHEN A FUNCTION IS PASSED AS A PARAMETER ITS CALLED A CALLBACK FUNCTION. ANONYMOUS FUNCTIONS ARE COMMON IN SUCH SCENARIOS.

Eg.
```
$arr = [1, 5, 7, 6, 3];

array_walk = ($arr, function($n){
                echo $n };);

// [1, 5, 7, 6, 3]
```

- ANONYMOUS FUNCTIONS ARE TERMINATED BY A SEMI COLON.

Eg.
```
$func = function ($a, $b) { return $a + $b; };

echo $func (2, 3); // 5
```

- MULTIPLE ANONYMOUS FUNCTIONS CAN BE STORED IN AN ARRAY AND BE ACCESSED VIA THE INDEX OF THE ARRAY.

- "CLOSURES" ARE ANONYMOUS FUNCTIONS THAT CAN ACCESS VARIABLES FROM SURROUNDING SCOPE, VIA "use" KEYWORD.

Eg.

```
$x = 10;
$y = 20;

$arr = [ function () use ($x, $y) {
            return $x * $y; },

        function () use ($x, $y) {
            return $x + $y; },
];

echo $arr[0]; // 200
echo $arr[1]; // 30
```

PROCESS	PRODUCT	TOTAL	SIGN
09	15	24.	

[EXPERIMENT -7]

(a) imagecreate()

— CREATES A NEW CANVAS/PALETTE BASED IMAGE

SYN function imagecreate(width, height)

width — SPECIFY WIDTH OF IMAGE
height — SPECIFY HEIGHT OF IMAGE

→ RETURNS A RESOURCE ON SUCESS OR FALSE

(b) imagecolorallocate()

— ALLOCATES A COLOR FOR AN IMAGE

SYN function imagecolorallocate(img, r, g, b)

img — SPECIFY IMAGE TO SET COLOR
r — RED VALUE
g — GREEN VALUE
b — BLUE VALUE

→ RETURNS A COLOR RESOURCE OR FALSE

(c) imagegif()

— OUTPUT IMAGE TO BROWSER OF FILE

SYN function imagegif(img, [file])

img — SPECIFY IMAGE
file — OPTIONAL, SPECIFY PATH TO SAVE FILE, ELSE RAW IMAGE STREAM IS SENT TO OUTPUT

→ RETURNS TRUE OR FALSE

④ imagejpeg()

— OUTPUT IMAGE TO BROWSER OR FILE

SYN function imagejpeg (img , [file], [quality])

img — SPECIFY IMAGE
file — SAME AS "imagegif()"
quality — OPTIONAL, RANGES FROM ZERO [WORST] TO 100 [BEST]

→ RETURNS TRUE OR FALSE

⑤ imagewbmp()

— OUTPUT IMAGE TO BROWSER OR FILE

SYN function imagewbmp (img, [file], [bg-color])

img — SPECIFY IMAGE
file — SAME AS "imagejpeg()"

bg - clr — SPECFY COLOR, OPTIONAL

→ RETURNS TRUE OR FALSE

(f) imagepng()

— OUTPUT A "PNG" IMAGE TO BRDWSR OR FILE

SYN function imagepng (img, [file], [quality], [filters])

img — SPECIFY IMAGE
file — SAME AS "imagewbmp"
quality — SAME AS "imagejpeg"
filters — OPTIONAL, USED FOR IMAGE SIZE COMPRESSION "PNG_NO_FILTER", "PNG_ALL_FILTERS"

→ RETURNS TRUE OR FALSE

(g) imagestring()

— DRAW A STRING HORIZONTALLY

SYN imagestring (img, font, x, y, string, color)

img — SPECIFY IMAGE
font — SET FONT
x — SET "x" COORDINATE
y — SET "y" COORDINATE
string — STRING TO DISPLAY
color — COLOR CREATED VIA "imagecolorallocate"

→ RETURNS TRUE OR FALSE

(b) imagecopyresized()

— COPY AND RESIZE PART OF AN IMAGE

SYN function imagecopyresized (

dst_img, src_img,
d_x , dy,
sx , sy,
dw , dh,
sw , sh
)

dst_img — DESTINATION IMAGE
src_img — SOURCE IMAGE
dx — "x" COORDINATE OF DESTINATION
dy — "y" COORDINATE OF DESTINATION
sx — "x" COORDINATE OF SOURCE
sy — "y" COORDINATE OF SOURCE
dw — DESTINATION WIDTH

dh — DESTINATION HEIGHT
sw — SOURCE WIDTH
sh — SOURCE HEIGHT

→ RETURNS TRUE OR FALSE

(1) imagecopyresampled ()

— COPY AND RESIZE PART OF AN IMAGE WITH RESAMPLING

SYN function imagecopyresampled (
d , s ,
dx , dy ,
sx , sy ,
dw , dh
sw , sh ')

d — SPECIFY DESTINATION IMAGE
s — SPECIFY SOURCE IMAGE
dx — "x" COORDINATE OF DESTINATION
dy — "y" COORDINATE OF DESTINATION
sx — "x" COORDINATE OF SOURCE
sy — "y" COORDINATE OF SOURCE
dw — DESTINATION WIDTH
dh — DESTINATION HEIGHT
sw — SOURCE WIDTH
sh — SOURE HEIGHT

→ RETURNS TRUE OR FALSE

(j) AddPage()

— USED TO ADD A PAGE TO "PDF" OBJECT
— STARTS FROM UPPER-LEFT BY 1cm FROM BORDER
— MARGINS CAN BE CHANGED VIA SETMARGIN()

SYN obj → AddPage()

obj — "PDF" OBJECT

(k) setFont()

— USED TO SPECIFY FONT

SYN obj → setFont(font, style, size)

obj — "PDF" OBJECT
font — SPECIFY FONT FAMILY
style — BOLD, ITALIC, UNDERLINE
size — SPECIFY SIZE

(l) Cell()

— USED TO CREATE A RECTACULAR AREA WHICH CONTAINS A LINE OF TEXT
— OUTPUT AT CURRENT LOCATION

SYN obj° → Cell (w, h, txt,
border, ln, align,
fill, link)

obj° → "PDF" OBJECT
w — CELL WIDTH. IF "0" THEN
CELL GOES TILL RIGHT MARGIN
h — CELL HEIGHT, DEFAULT VALUE IS 0
txt — STRING TO PRINT, DEFAULT IS
AN EMPTY STRING
border — SPECIFY BORDER, 0 - NO
BORDER, 1 - FRAME,
L, T, R, B
ln — WHERE CURSOR GOES AFTER
CALL. 0 - RIGHT, 1 - START
OF NEXT LINE, 2 - BELOW
align — ALIGN TEXT, L, R, C
fill — IF CELL BACKGROUND BE
PAINTED - (BOOLEAN) - DEFAULT → F
link — URL OR IDENTIFIER RETURNED
BY AddLink()

PROCESS	PRODUCT	TOTAL	SIGN
09	15	24.	