**rand:**

The rand() function generates a random integer.

**Tip:** If you want a random integer between 10 and 100 (inclusive), use rand (10,100).

**Tip:** The mt_rand() function produces a better random value, and is 4 times faster than rand().

Format : rand(); or rand(*min*,*max*);

**substr :**

The substr() function returns a part of a string.

substr(*string*,*start*,*length*);

| Parameter | Description |
|---|---|
| *string* | Required. Specifies the string to return a part of |
| *start* | Required. Specifies where to start in the string<br>• A positive number - Start at a specified position in the string<br>• A negative number - Start at a specified position from the end of the string<br>• 0 - Start at the first character in string |
| *length* | Optional. Specifies the length of the returned string. Default is to the end of the string.<br>• A positive number - The length to be returned from the start parameter<br>• Negative number - The length to be returned from the end of the string |

**Note:** If the start parameter is a negative number and length is less than or equal to start, length becomes 0.

**str_repeat :**

The str_repeat() function repeats a string a specified number of times.
str_repeat(*string*,*repeat*)
str_repeat(“hi”,*5*) ; //It will print 5 times hi.

**str_shuffle :**

The str_shuffle() function randomly shuffles all the characters of a string.
str_shuffle(*string*);
str_shuffle(*“Hello World”*); // Output may be : l HroedlloW

**Imagefontwidth:**

int **imagefontwidth** ( int `$font` )
Returns the pixel width of a character in font.

**imagecreatetruecolor:**

**imagecreatetruecolor()** returns an image identifier representing a black image of the specified. size.
resource **imagecreatetruecolor** ( int `$width` , int `$height` );

**ImageColorAllocate:**

Returns a color identifier representing the color composed of the given RGB components.

int **imagecolorallocate** ( resource `$image` , int `$red` , int `$green` , int `$blue` )

**Note:** The first call to **imagecolorallocate()** fills the background color in palette-based images.

**imagecolorallocate()** must be called to create each color that is to be used in the image represented by `image`.

**Imagefill:**

bool **imagefill** ( resource `$image` , int `$x` , int `$y` , int `$color` )
Performs a flood fill starting at the given coordinate (top left is 0, 0) with the given `color` in the `image`.

**Imagestring:**

bool **imagestring** ( resource `$image` , int `$font` , int `$x` , int `$y` , string `$string` , int `$color` )
Draws a `string` at the given coordinates.

| **Parameter** | **Description** |
|---|---|
| image | An image resource, returned by one of the image creation functions, such as imagecreatetruecolor(). |
| font | Can be 1, 2, 3, 4, 5 for built-in fonts in latin2 encoding (where higher numbers corresponding to larger fonts) or any of your own font identifiers registered with imageloadfont(). |
| X , Y | X and Y coordinates of the upper left corner. |
| string | The string to be written. |
| color | A color identifier created with imagecolorallocate(). |

**ImagePng:**

imagepng — Output a PNG image to either the browser or a file.

bool **imagepng** ( resource `$image` [, string `$filename` [, int `$quality` [, int `$filters` ]]] )

## ImageCreate:

**resource imagecreate ( int `$width` , int `$height` )**

**imagecreate()** returns an image identifier representing a blank image of specified size.

In general, we recommend the use of imagecreatetruecolor() instead of **imagecreate()** so that image processing occurs on the highest quality image possible.

## ImageTTFText:

Writes the given `text` into the image using TrueType fonts.

array **imagettftext** ( resource `$image` , float `$size` , float `$angle` , int `$x` , int `$y` , int `$color` , string `$fontfile` , string `$text` )

| **Parameter** | **Description** |
|---|---|
| image | An image resource, returned by one of the image creation functions, such as imagecreatetruecolor(). |
| size | The font size. This should be specified as the pixel size. |
| angle | The angle in degrees, with 0 degrees being left-to-right reading text. Higher values represent a counter-clockwise rotation. For example, a value of 90 would result in bottom-to-top reading text. |
| X and Y | The coordinates given by `x` and `y` will define the basepoint of the first character (roughly the lower-left corner of the character). This is different from the imagestring(), where `x` and `y` define the upper-left corner of the first character. For example, "top left" is 0, 0. |
| color | The color index. Using the negative of a color index has the effect of turning off antialiasing. See imagecolorallocate(). |
| fontfile | The path to the TrueType font you wish to use. |
| text | Data that you wants to display |

## ImageDestroy:

frees any memory associated with image `image`.

**bool imagedestroy ( resource `$image` )**

**- Prof. Tushar V. Ratanpara**
**- Prof. Apurva A. Mehta**
**- Prof. Hariom A. Pandya**