1. Alert Box

An alert box is often used if you want to make sure information comes through to the user.

When an alert box pops up, the user will have to click "OK" to proceed.

Syntax

window.alert("*sometext*");

The window.alert() method can be written without the window prefix.

Example

alert("I am an alert box!");

2. Confirm Box

A confirm box is often used if you want the user to verify or accept something.

When a confirm box pops up, the user will have to click either "OK" or "Cancel" to proceed.

If the user clicks "OK", the box returns **true**. If the user clicks "Cancel", the box returns **false**.

Syntax

window.confirm("*sometext*");

The window.confirm() method can be written without the window prefix.

Example

```
if (confirm("Press a button!")) {
  txt = "You pressed OK!";
} else {
  txt = "You pressed Cancel!";
}
```

3. Prompt Box

A prompt box is often used if you want the user to input a value before entering a page.

When a prompt box pops up, the user will have to click either "OK" or "Cancel" to proceed after entering an input value.

If the user clicks "OK" the box returns the input value. If the user clicks "Cancel" the box returns null.

Syntax

window.prompt("*sometext*","*defaultText*");

The window.prompt() method can be written without the window prefix.

Example

```
let person = prompt("Please enter your name", "Harry Potter");
let text;
if (person == null || person == "") {
  text = "User cancelled the prompt.";
} else {
  text = "Hello " + person + "! How are you today?";
}
```

4. Line Breaks

To display line breaks inside a popup box, use a back-slash followed by the character n.

Example

```
alert("Hello\nHow are you?");
```

5. Window alert()

Example

Display an alert box:

```
alert("Hello! I am an alert box!!");
```

The alert() method displays an alert box with a message and an OK button.

The alert() method is used when you want information to come through to the user.

CSS Functions

acos()	Returns the inverse cosine of a number between -1 and 1
asin()	Returns the inverse sine of a number between -1 and 1
atan()	Returns the inverse tangent of a number between -∞ and ∞
atan2()	Returns the inverse tangent of two values between -infinity and infinity
attr()	Returns the value of an attribute of the selected element
blur()	Applies a blur effect to an element
brightness()	Adjusts the brightness of an element (brighter or darker)
calc()	Allows you to perform calculations to determine CSS property values
circle()	Defines a circle
clamp()	Sets a value that will adjust responsively between a minimum value and a maximum value depending on the size of the viewport
color()	Allows a color to be specified in a particular, specified color space
color-mix()	Mixes two color values in a given color space, by a given amount
conic-gradient()	Creates a conic gradient
contrast()	Adjusts the contrast of an element
cos()	Returns the cosine of an angle
counter()	Returns the current value of the named counter

counters()	Returns the current values of the named and nested counters
cubic-bezier()	Defines a Cubic Bezier curve
drop-shadow()	Applies a drop shadow effect to an image
ellipse()	Defines an ellipse
exp()	Returns E raised to the power of the specified number x (E^x)
fit-content()	Allows you to size an element based on its content
grayscale()	Converts an image to grayscale
hsl() / hsla()	Defines a color using the Hue-Saturation-Lightness model (HSL); with an optional alpha component
hue-rotate()	Applies a color rotation to an element
hwb()	Defines a color using the Hue-Whiteness-Blackness model (HWB); with an optional alpha component
hypot()	Returns the square root of the sum of squares of its parameters
inset()	Defines a rectangle at the specified inset distances from each side of the reference box
invert()	Inverts the color of an image
lab()	Specifies a color in the CIE L*a*b color space
lch()	Specifies a color in the LCH (Lightness-Chroma-Hue) color space
light-dark()	Enables two color-value settings, and returns the first value if the user has set a light color theme, and the second value if the user has set a dark color theme

linear-gradient()	Creates a linear gradient
log()	Returns the natural logarithm (base E) of a specified number, or the logarithm of the number to the specified base
matrix()	Defines a 2D transformation, using a matrix of six values
matrix3d()	Defines a 3D transformation, using a 4x4 matrix of 16 values
max()	Uses the largest value, from a comma-separated list of values, as the property value
min()	Uses the smallest value, from a comma-separated list of values, as the property value
minmax()	Defines a size range greater than or equal to a *min* value and less than or equal to a *max* value (used with CSS grids)
mod()	Returns the remainder left over when a number is divided by another number
oklab()	Specifies a color in the OKLAB color space
oklch()	Specifies a color in the OKLCH color space
opacity()	Applies an opacity effect to an element
perspective()	Defines a perspective view for a 3D transformed element
polygon()	Defines a polygon
pow()	Returns the value of a number (x) raised to the power of another number (y)
radial-gradient()	Creates a radial gradient
ray()	Defines the offset-path line segment that an animated element should follow

rem()	Returns the remainder left over when a number is divided by another number
repeat()	Repeats a set of columns or rows in a grid
repeating-conic-gradient()	Repeats a conic gradient
repeating-linear-gradient()	Repeats a linear gradient
repeating-radial-gradient()	Repeats a radial gradient
rgb() / rgba()	Defines colors using the Red-Green-Blue model (RGB); with an optional alpha component
rotate()	Defines a 2D rotation of an element
rotate3d()	Defines a 3D rotation of an element
rotateX()	Defines a 3D rotation of an element around the x-axis (horizontal)
rotateY()	Defines a 3D rotation of an element around the y-axis (vertical)
rotateZ()	Defines a 3D rotation of an element around the z-axis
round()	Rounds a number according to the specified rounding-strategy
saturate()	Adjusts the saturation (color intensity) of an element
scale()	Defines a 2D scaling of an element
scale3d()	Defines a 3D scaling of an element
scaleX()	Scales an element horizontally (the width)

scaleY()	Scales an element vertically (the height)
sepia()	Converts an image to sepia
sin()	Returns the sine of a number (angle)
skew()	Skews an element along the x- and y-axis
skewX()	Skews an element along the x-axis
skewY()	Skews an element along the y-axis
sqrt()	Returns the square root of a number
steps()	Creates a stepped timing function for animations
tan()	Returns the tangent of a number
translate()	Allows you to re-position an element along the x- and y-axis
translateX()	Allows you to re-position an element along the x-axis
translateY()	Allows you to re-position an element along the y-axis
url()	Allows you to include a file in the stylesheet
var()	Inserts the value of a custom property