Chapter 44: Modals - Prompts

Section 44.1: About User Prompts

User Prompts are methods part of the Web Application API used to invoke Browser modals requesting a user action such as confirmation or input.

`window.alert(message)`

Show a modal *popup* with a message to the user.
Requires the user to click [OK] to dismiss.

```
alert("Hello World");
```

More information below in "Using alert()".

`boolean = window.confirm(message)`

Show a modal *popup* with the provided message.
Provides [OK] and [Cancel] buttons which will respond with a boolean value `true` / `false` respectively.

```
confirm("Delete this comment?");
```

`result = window.prompt(message, defaultValue)`

Show a modal *popup* with the provided message and an input field with an optional pre-filled value.
Returns as `result` the user provided input value.

```
prompt("Enter your website address", "http://");
```

More information below in "Usage of prompt()".

`window.print()`

Opens a modal with document print options.

```
print();
```

Section 44.2: Persistent Prompt Modal

When using **prompt** a user can always click ***Cancel*** and no value will be returned.
To prevent empty values and make it more **persistent**:

```
<h2>Welcome <span id="name"></span>!</h2>
```

```
<script>
// Persistent Prompt modal
var userName;
while(!userName) {
  userName = prompt("Enter your name", "");
  if(!userName) {
    alert("Please, we need your name!");
  } else {
    document.getElementById("name").innerHTML = userName;
  }
}
</script>
```

jsFiddle demo

Section 44.3: Confirm to Delete element

A way to use `confirm()` is when some UI action does some *destructive* changes to the page and is better accompanied by a **notification** and a **user confirmation** - like i.e. before deleting a post message:

```
<div id="post-102">
  <p>I like Confirm modals.</p>
  <a data-deletepost="post-102">Delete post</a>
</div>
<div id="post-103">
  <p>That's way too cool!</p>
  <a data-deletepost="post-103">Delete post</a>
</div>
```

```
// Collect all buttons
var deleteBtn = document.querySelectorAll("[data-deletepost]");

function deleteParentPost(event) {
  event.preventDefault(); // Prevent page scroll jump on anchor click

  if( confirm("Really Delete this post?") ) {
    var post = document.getElementById( this.dataset.deletepost );
    post.parentNode.removeChild(post);
    // TODO: remove that post from database
  } // else, do nothing

}

// Assign click event to buttons
[].forEach.call(deleteBtn, function(btn) {
  btn.addEventListener("click", deleteParentPost, false);
});
```

jsFiddle demo

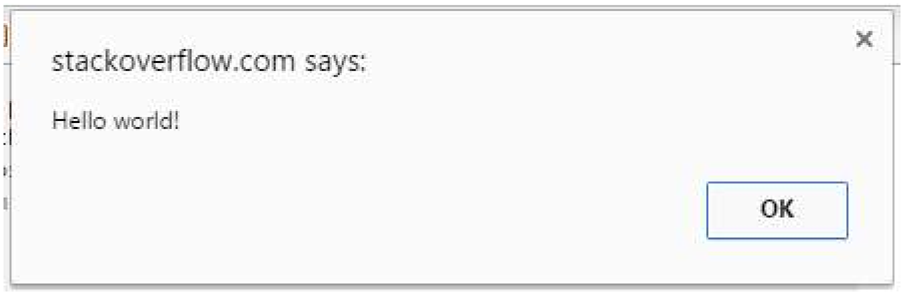
Section 44.4: Usage of alert()

The `alert()` method of the `window` object displays an *alert box* with a specified message and an `OK` or `Cancel` button. The text of that button depends on the browser and can't be modified.

Syntax

```
alert("Hello world!");
// Or, alternatively...
window.alert("Hello world!");
```

Produces


stackoverflow.com says:

Hello world!

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

Note: The alert box takes the focus away from the current window, and forces the browser to read the message. Do not overuse this method, as it prevents the user from accessing other parts of the page until the box is closed. Also it stops the further code execution, until user clicks `OK`. (in particular, the timers which were set with `setInterval()` or `setTimeout()` don't tick either). The alert box only works in browsers, and its design cannot be modified.

Parameter	Description
message	Required. Specifies the text to display in the alert box, or an object converted into a string and displayed.

Return value

`alert` function doesn't return any value

Section 44.5: Usage of prompt()

Prompt will display a dialog to the user requesting their input. You can provide a message that will be placed above the text field. The return value is a string representing the input provided by the user.

```
var name = prompt("What's your name?");
console.log("Hello, " + name);
```

You can also pass `prompt()` a second parameter, which will be displayed as the default text in the prompt's text field.

```
var name = prompt('What\'s your name?', ' Name...');
console.log('Hello, ' + name);
```

Parameter	Description
message	Required. Text to display above the text field of the prompt.
default	Optional. Default text to display in the text field when the prompt is displayed.