

JavaScript Events

User Interface events (load)

Focus & Blur

Mouse Events

Keyboard events

Form events

Mutation events

HTML5 events

Mouse Events

click	The event occurs when the user clicks on an element
dblclick	The event occurs when the user double-clicks on an element
mousedown	The event occurs when a user presses a mouse button over an element
mouseover	The event occurs when the pointer is moved onto an element, or onto one of its children
mouseout	The event occurs when a user moves the mouse pointer out of an element, or out of one of its children
mouseup	The event occurs when a user releases a mouse button over an element
mousemove	occurs when the cursor is moved around the element

Mouse Events

Example: click.html

Keyboard Events

- input Occurs when the value of <input> or <textarea> changes
- keydown The event occurs when the user is pressing a key
- keypress The event occurs when the user presses a key
- keyup The event occurs when the user releases a key

Keyboard Events

Example: keypress.html

Event listener checks for keypress event on the <textarea> element
Textfiled is limited to 180 characters

Each time it fires, the charCount() function updates the character count and shows the last character used.

Each key is shown in ASCII code

HTML ASCII Code

ASCII = American Standard Code for Information Interchange

It was designed in the early 60's, as a standard character set for computers and electronic devices.

ASCII is a 7-bit character set containing 128 characters.

It contains the numbers from 0-9, the upper and lower case English letters from A to Z, and some special characters.

The character sets used in modern computers, in HTML, and on the Internet, are all based on ASCII.

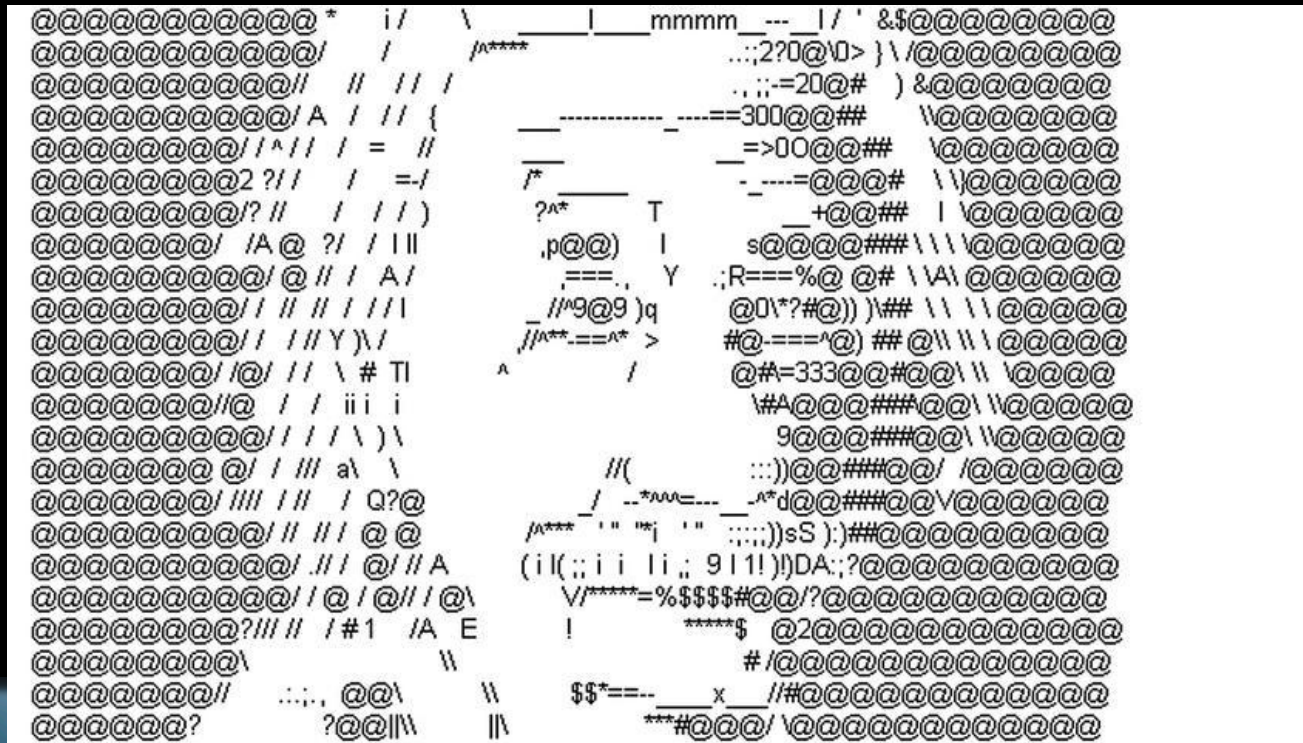
HTML ASCII Code

Char	Number	Description
A	65	uppercase A
B	66	uppercase B
C	67	uppercase C
D	68	uppercase D
E	69	uppercase E
F	70	uppercase F
G	71	uppercase G
H	72	uppercase H

[http://www.w3schools.com/Charsets/ref_html_ascii.asp](http://www.w3schools.comCharsets/ref_html_ascii.asp)

ASCII Art

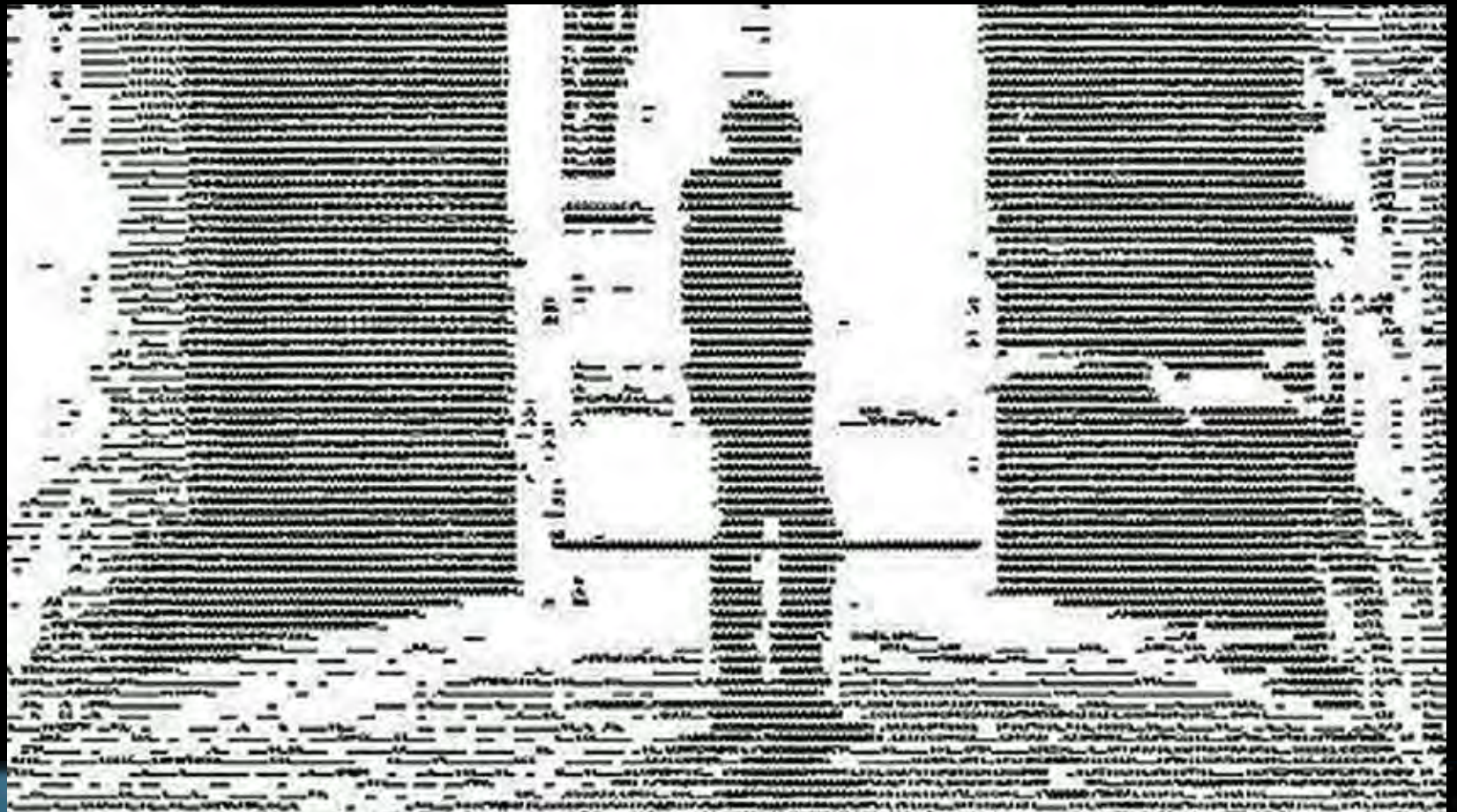
ASCII art is an early graphic-design technique, dating back to the 1890s when typewriters became more than just a new tool for writing.



ASCII Art



ASCII Art



Keyboard Events

```
var el;  
function charCount(e) {  
  var textEntered, charDisplay, counter, lastkey;  
  textEntered = document.getElementById('message').value;  
  charDisplay = document.getElementById('charactersLeft');  
  counter = (180 - (textEntered.length));  
  charDisplay.textContent = counter;  
  
  lastkey = document.getElementById('lastKey');  
  lastkey.textContent = 'Last key in ASCII code: ' + e.keyCode; }  
  
el = document.getElementById('message');  
el.addEventListener('keypress', charCount, false);
```

Form Events

submit	Occurs when form is submitted
change	occurs when the status of forms change (ex. radio button selected)
input	occurs when user types in text in the <code><input></code> or <code><textarea></code>

Form Events

Example: form.html

The change events triggers the PackageHint() function.
This shows different messages below the menu to reflect the choice.

Form Events

Exercise: modify the form.html and javascript to offer the user a drop down menu with two choices : 20 or younger / 21 or older.
When the use interacts with the drop-down menu, the messages below say “ You can drink” or “You cannot drink”

Mutation Events

When elements are added to or removed from the DOM, its structure changes.

These changes trigger mutation events.

Mutation observers are designed to wait until a script has finished its task before reacting, then report the changes as a batch (rather than one at a time).

Mutation Events

DOMNodeInserted

Fires when a node is inserted into the DOM tree e.g. using `appendChild()`, `replaceChild()`, or `insertBefore()`

DOMNodeRemoved

Fires when a node is removed from the DOM tree e.g. `removeChild()`

DOMSubtreeModified

Fires when the DOM structure changes after the two events listed above occur.

Mutation Events

DOMNodeInsertedIntoDocument

Fires when a node is inserted into the DOM tree as a descendant of another node that is already in the document.

DOMNodeRemovedFromDocument

Fires when a node is removed into the DOM tree as a descendant of another node that is already in the document.

Mutation Events

Example: mutation.html

Two event listeners each trigger their own function.

The first listens for when the user clicks the link to add a new list item. It then uses DOM manipulation events to add a new element (changing the DOM structure and triggering mutation events).

The second event listener waits for the DOM tree within the `` element to change. When the `DOMNodeInserted` event fires, it calls a function `updateCount()`. This function counts how many items there are in the list, and then updates the list count at the top of the page accordingly.

HTML5 Events

DOMContentLoaded

Fires when the DOM tree is formed (images, css and js might be still loading). Script starts to run earlier than using load event which waits for other resources such as images to load.

hashchange

Fires when the URL hash changes (w/o the entire window refreshing). Hashes are used on links to specific parts (known as anchors) within a page and also on pages that use AJAX to load content.

HTML5 Events

Anchor link

```
<a id="tips">Useful Tips Section</a>
```

```
<a href="#tips">Visit the Useful Tips Section</a>
```

HTML5 Events

beforeunload

Fires on the window object before the page is unloaded. It should only be used to help the user (not to encourage them to stay on the website if they are trying to leave). Example. It could be helpful to let a user know that changes on a form he competed have not been saved.

HTML5 Events

Example: `html5_events.html`

As soon as the DOM tree has been formed the focus is given to the text input.

The `DOMContentLoaded` event fires before the load event.

If users try to leave the page before they press submit button, the `beforeunload` event checks that they want to leave.

HTML5 Events

Example.html

Shows an interface for a user to record voice notes.

The user can enter a name which is displayed in the heading, and they can press record (which changes the image that is shown).

Keyup event triggers a function writeLabel() which copies the text from the form input and writes it into the main heading.

The record button has an attribute data-state. When the page loads, its value is record. When user presses the button, the value of this attribute changes to pause (and triggers a new css rule to indicate that it is now recording).

JavaScript Events

Events are the browser's way of indicating when something has happened (e.g. a page has finished loading, button has been clicked)

When an event occurs on an element, it can trigger a JavaScript function. When this function then changes the web page in some way, it feels interactive because it has responded to the user.

You can use event delegation to monitor for events that happen on all of the children of an element.

The most commonly used events are W3C DOM events, although there are others in the HTML5 specifications as well as browser-specific events.

JavaScript Events

Assignment: JavaScript events.

Write a script that will calculate a 15% and a 20% tip.

The users type into a text field the amount that they wish to tip on.

The user selects 15% or 20% they wish to tip via a drop-down menu.

When they click on a "Calculate" button the amount of the tip and the total amount (tip plus original amount) are displayed.