

WEB230: JavaScript 1

Module 6: Handling Events Part B - Events

Script Execution Timeline

- no two scripts can run at the same time
- each peice of code (often functions) will wait for others to finish
- web workers provide a way to do something while other things run

The event Objects

- event handlers can accept a parameter called the event object
- this object has information about the event
 - for example, which element was clicked on
 - which button or key was pressed
- properties and methods vary depending on the type of event
- this parameter is usually called `event` or simply `e`

Mouse Clicks

- `mousedown`, `mouseup`, `click`, and `dblclick` events
- if you need the location of the mouse click you can use `event.clientX` and `event.clientY` properties
- `event.button` takes into account user customization
 - 0: Main button pressed, usually the left button or the un-initialized state
 - 1: Auxiliary button pressed, usually the wheel button or the middle button (if present)
 - 2: Secondary button pressed, usually the right button
 - 3: Fourth button, typically the Browser Back button
 - 4: Fifth button, typically the Browser Forward button

Mouse Button Event Order

1. `mousedown`
2. `mouseup`
3. `click`
4. `dblclick` - if applicable
 - `dblclick` will repeat the previous three twice

Mouse Motion

- `mousemove` event every time the mouse moves
- `mouseover` or `mouseout` event equivalent to CSS `:hover`
- there are also `drag` events when moving objects

Keyboard Events

- `keydown` and `keyup` events

- `keydown` will repeat if held
- `event.key` holds a string with the value that the key would type
- boolean properties for modifier keys:
 - `event.shiftKey`
 - `event.ctrlKey`
 - `event.altKey`
 - `event.metaKey` (Windows key or Mac Command key)
- event occurs on element that has focus (or `document.body`)
- if you want to capture all keystrokes, use `window.addEventListener()`
 - `window.` is optional since it is the global object
- Note: the `keypress` event is deprecated

Key Event Properties

- `event.key` (String) The key value of the key represented by the event. If the value has a printed representation, this attribute's value is the same as the `char` attribute. Otherwise, it describes the key.
- `event.code` (String) Holds a string that identifies the physical key being pressed. The value is not affected by the current keyboard layout or modifier state, so a particular key will always return the same value.

```
document.body.addEventListener('keydown', function(event) {
  console.log('Key pressed:', event.key);
});
```

- `event.repeat` (Boolean) `true` if the key is being held down such that it is automatically repeating
 - can be used to avoid repeatedly running the event handler

```
document.body.addEventListener('keydown', function(event) {
  if (event.repeat) { return; }
  console.log('Key pressed:', event.key);
});
```

Scroll Events

- `scroll` event when page scrolls
- fired every time the page is scrolled
- `window.scrollX` and `window.scrollY` for scroll position

Focus Events

- `focus` and `blur`
- when an element is selected it has `focus`
- when it loses focus a `blur` event is fired
- most often used with forms
- does not propagate

Setting Timers

- `setTimeout` to run a function after an amount of time
- schedules a function to be called in a specified amount of time
- `clearTimeout` can be used to cancel it
- `setInterval` and `clearInterval` is similar but repeats every specified time interval

```
const button = document.querySelector('button');
const list = document.querySelector('ul');
let interval;
button.addEventListener('click', function(event){
  if(interval) {
    clearInterval(interval);
  } else {
    interval = setInterval(function(){
      let item = document.createElement('li');
      item.textContent = 'New item';
      list.appendChild(item);
    }, 1000);
  }
});
```