Генерация событий

Генерация события

Event

```
event = new Event(typeArg, eventInit);
```

- Тип события – может быть как своим, так и встроенным, к примеру "click"
- Флаги – объект вида { bubbles: true/false, cancelable: true/false }...

```
1 var ev = new Event("look", {"bubbles":true, "cancelable":false});
2 document.dispatchEvent(ev);
```

Генерация события

dispatchEvent

- Отправляет событие в общую систему событий
- Событие подчиняется тем же правилам поведения (захват и всплытие) как и непосредственно инициированные события

```
cancelled = !target.dispatchEvent(event)
//event - объект события, который инициализируется
//target - используется для инициализации Event.target
//и установки события, которое обработчик вызывает

//return - false если хотя бы один из обработчиков
//этого события вызовал preventDefault
```

```
<button id="elem" onclick="alert('Клик');">Автоклик</button>
<script>
  var event = new Event("click");
  elem.dispatchEvent(event);
</script>
```

Генерация события

CustomEvent

- идентичен Event(), НО у второго аргумента-объекта есть дополнительное свойство detail, в котором можно указывать дополнительную информацию

```
<h1 id="elem">Element</h1>

<script>
var elem = document.getElementById('elem');
  elem.addEventListener("hello", function(event) {
    console.log( event.detail.name );
  }, false);

  var event = new CustomEvent("hello", {
    detail: { name: "something" }
  });

  elem.dispatchEvent(event);
</script>
```

Таймеры

Таймеры в js

Таймеры - это не sleep(), они создают события, которые используют Event Loop

- setTimeout(function, timeout) - не ранее чем через timeout
- setInterval(function, timeout) - не чаще чем через timeout

Таймеры в js

```
function printMessage()
 { alert(‘Hello'); }
setTimeout(sayHi, 1000);   // функция отработает с задержкой в 1000ms

var foo = setInterval(() => console.log(‘Hello'), 1500); // функция отрабатывает циклически каждые 1500ms
```