

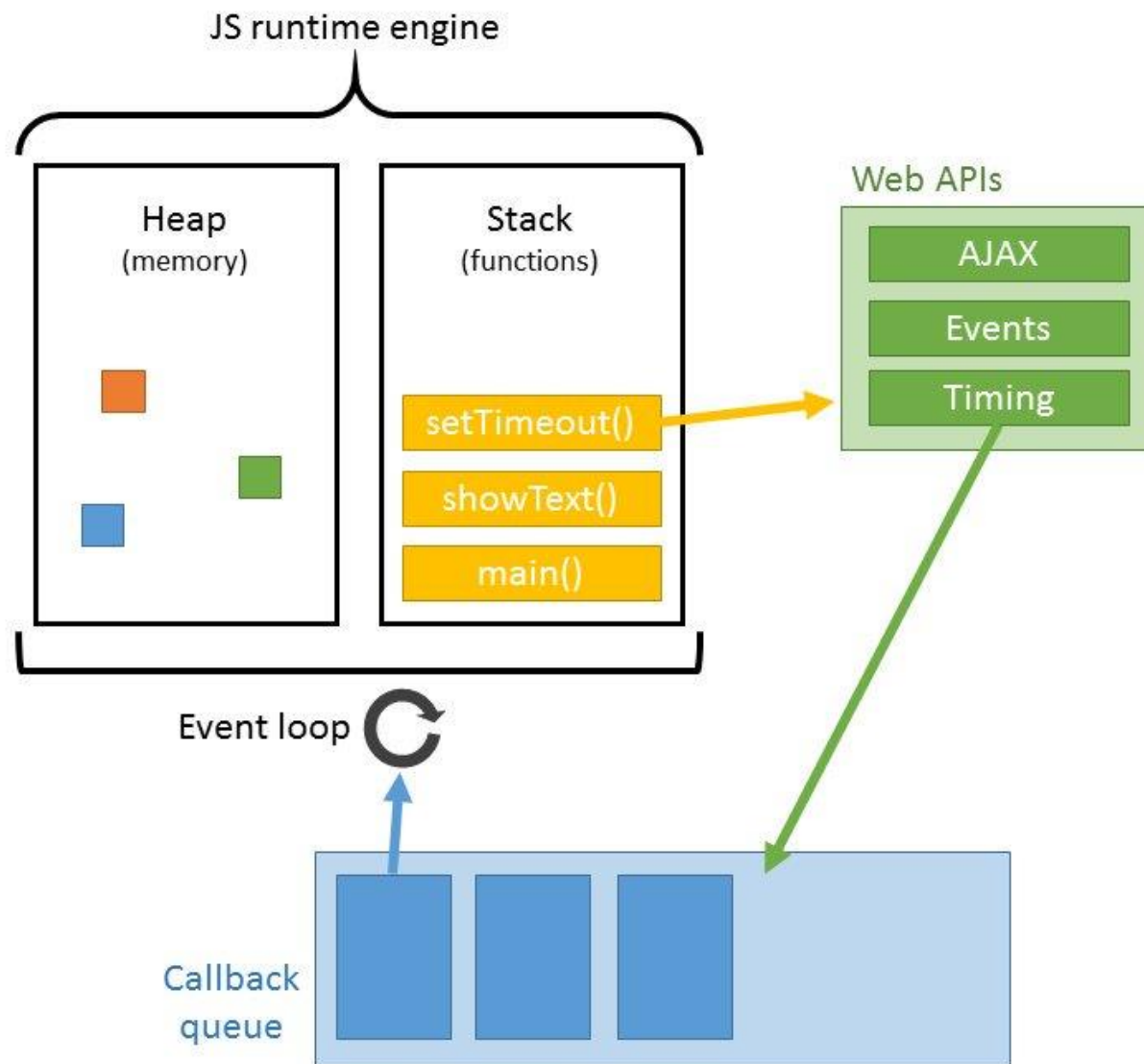
Events, JQuery, JS Frameworks overview

Событие

- Событие — это некоторое действие, которое было совершено в прошлом, пользователем, и на которое разработчик может обработать с необходимой заложенной логикой. Например: клик по кнопке, перетаскивание, двойной клик, и т.д.



Очередь событий



Разновидности событий

События мыши:

- **click** – происходит, когда кликнули на элемент левой кнопкой мыши
- **contextmenu** – происходит, когда кликнули на элемент правой кнопкой мыши
- **mouseover** – возникает, когда на элемент наводится мышь
- **mousedown** и **mouseup** – когда кнопку мыши нажали или отжали
- **mousemove** – при движении мыши

События на элементах управления:

- **submit** – посетитель отправил форму `<form>`
- **focus** – посетитель фокусируется на элементе, например нажимает на `<input>`

Клавиатурные события:

- **keydown** – когда посетитель нажимает клавишу
- **keyup** – когда посетитель отпускает клавишу

События документа:

- **DOMContentLoaded** – когда HTML загружен и обработан, DOM документа полностью построен и доступен.

События CSS:

- **transitionend** – когда CSS-анимация завершена.

DOM модели

- DOM Level 0
- DOM Level 2

DOM Level 0

```
<p onclick="handler()"></p>

<script>
    function handler(){
        // Your code
    }
</script>
```

```
<p id="test"></p>
<script>
    var el = document.getElementById("test");
    el.onclick= function(){ /* Your code*/}
</script>
```

DOM Level 2

- **event**—имя события. Без приставки “on”, например “click”, “load”, именование с приставкой on “onclick”, “onload” не допускается
- **handler**—функция-обработчик
- **capture**—true-обработчик срабатывает на этапе перехвата, false —обработчик срабатывает на этапе всплытия.

```
<p id="test"></p>

<script>
  var el = document.getElementById("test");
  var event = "click";
  var handler = function(){ /* Your code*/}
  var capture = true;
  el.addEventListener(event, handler, capture);
</script>
```

DOM Level 2 for IE ☺

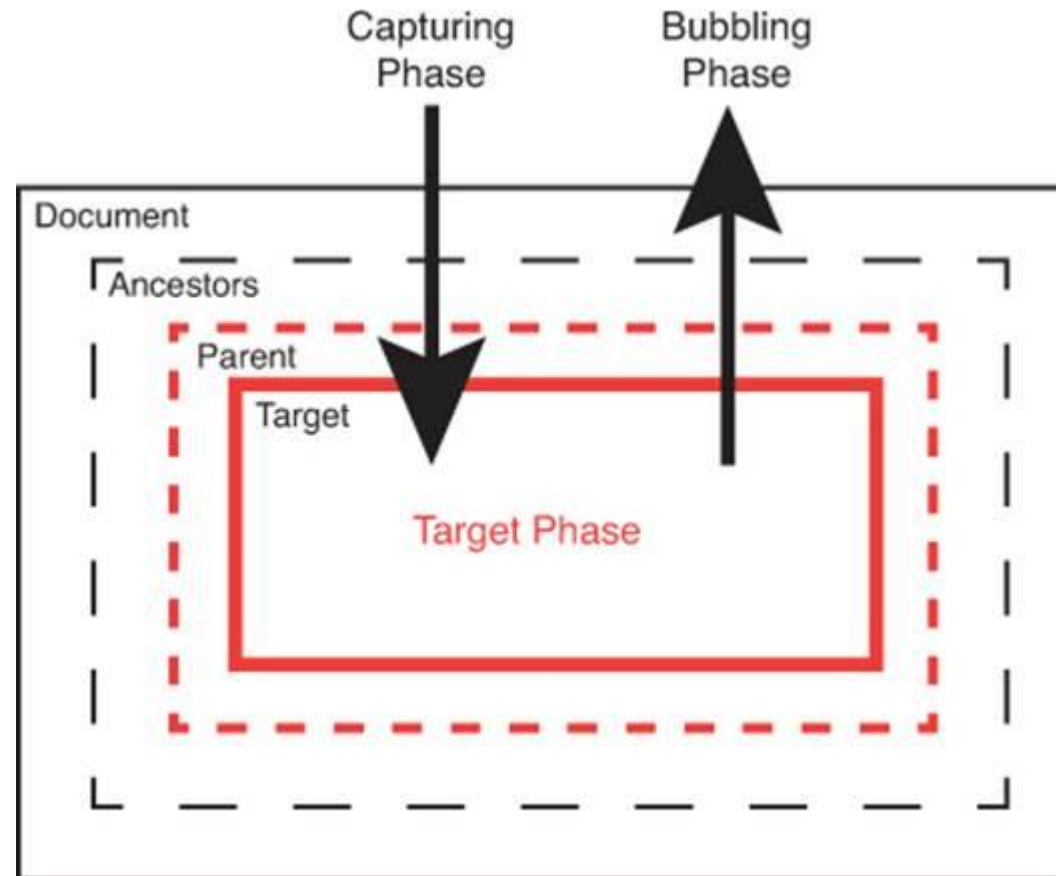
- **event** – имя события с приставкой “on”, например “onclick”
- **handler**–функция-обработчик
- **capture**–true.

```
<p id="test"></p>
```

```
<script>  
    var el = document.getElementById("test");  
    var event = "click";  
    var handler = function(){ /* Your code*/}  
    el.attachEvent(event, handler);  
</script>
```


Фазы событий

- capturing
- bubbling
- target



Удаление обработчика события

- `removeEventListener(event, handler, capture)` - удаление обработчика события, подписанного с помощью `addEventListener`;
- `detachEvent(event, handler)` - удаление обработчика события подписанного с помощью `attachEvent`. **Работает только для IE.**

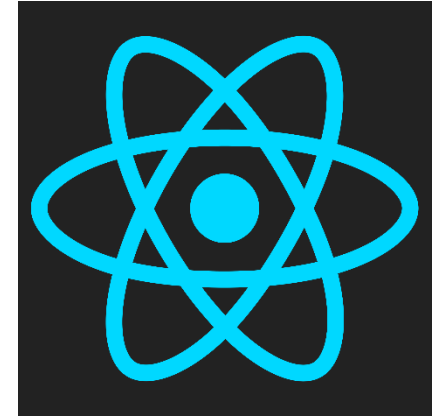
Объект Event

- **type** – тип события, например click
- **target** – элемент, на котором сработал обработчик.
- **currentTarget** – узел в котором в данный момент обрабатывается событие.
- **eventPhase** – 1 – этап перехвата, 2 - этап обработки в целевом элементе, 3 - этап всплытия
- **timestamp** – дата и время, когда произошло событие.
- **Bubbles** – всплывает ли данное событие.
- **Cancelable** – связано ли событие с действием по умолчанию.
- **clientX / clientY** – координаты курсора в момент клика (относительно окна)
- **stopPropagation()** – прекращение распространения события вверх по дереву.
- **preventDefault()** – отмена стандартного поведения браузера (действия по умолчанию).

Объект MouseEvent

- **button** – число, указывающее какая кнопка мыши изменила свое состояние при событии.
- **altKey, ctrlKey, shiftKey** – указывает на состояние соответствующей клавиши.
- **clientX, clientY** – координаты относительно окна браузера.
- **screenX, screenY** – координаты относительно экрана.

JS Libraries and Frameworks



JQuery

jQuery is a **fast, small, and feature-rich JavaScript library**. It makes **things** like HTML document traversal and manipulation, event handling, animation, and Ajax much simpler with **an easy-to-use** API that **works across a multitude of browsers**.

События в JQuery

В JQuery аналогом `addEventListener()` есть функция **`on()`**. Она более универсальна и кроссбраузерна. **`off()`** –аналог `removeEventListener`.

Также есть возможность прямого добавления события к элементу с помощью методов событий:

```
<button id="button">Click me</button>
```

```
<script>
```

```
    $("#button").on("click", function () { ... } );
```

```
    $("#button").click(function () { ... } );
```

```
</script>
```

События в JQuery

| <code>.on()</code> | Функция добавляет обработчик на указанное событие к элементу или набору элементов |
|-------------------------|--|
| <code>.off()</code> | Функция удаляет обработчик события с выбранного элемента |
| <code>.one()</code> | Функция добавляет обработчик на указанное событие к элементу или набору элементов,но вызывается только ОДИН раз. |
| <code>.trigger()</code> | Функция позволяет запустить событие другого элемента управления на странице. |

Q&A