Функции, события

JavaScript

Функции JavaScript

```
function sum(a, b) {
  return a + b;
}

let result = sum(1, 2);
alert( result ); // 3

// a и b – параметры, a+b – возвращаемый результат
```

События JavaScript

События мыши:

- **click** – происходит, когда кликнули на элемент левой кнопкой мыши (на устройствах с сенсорными экранами оно происходит при касании).
- **contextmenu** – происходит, когда кликнули на элемент правой кнопкой мыши.
- **mouseover / mouseout** – когда мышь наводится на / покидает элемент.
- **mousedown / mouseup** – когда нажали / отжали кнопку мыши на элементе.
- **mousemove** – при движении мыши.

События JavaScript

Клавиатурные события:

- **keydown и keyup** – когда пользователь нажимает / отпускает клавишу.

События документа:

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

CSS events:

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

Вызов события

1) **Вызов через атрибут:**

```
<input type="button" id="elem" onclick="alert('Было')" value="
Нажми меня">
```

2) Назначение обработчика в JS

```
button.onclick = sayThanks;
```

3) Использования this

```
<button onclick="alert(this.innerHTML)">Нажми
меня</button>
```

Вызов события

4) Использования метода addEventListener

```
elem.addEventListener("click", handler1); // Спасибо!
elem.addEventListener("click", handler2);
```

// **click** – событие **handler1, handler2** – функции

5) Отложенный вызов функций

let timerId = setTimeout(func|code, [delay], [arg1], [arg2], ...)

Параметры:

func|code - Функция или строка кода для выполнения. Обычно это функция. По историческим причинам можно передать и строку кода, но это не рекомендуется.

delay - Задержка перед запуском в миллисекундах (1000 мс = 1 с). Значение по умолчанию – 0.

arg1, arg2… - Аргументы, передаваемые в функцию

Вызов события

6) Отмена отложенной функции - clearTimeout

```
let timerId = setTimeout(...);
clearTimeout(timerId);
```

7) Вызов функции через интервалы времени

```
let timerId = setInterval(func|code, [delay], [arg1], [arg2], ...)
```

8) Удаление периодического запуска функции

```
let timerId = setTimeout(...); clearTimeout(timerId);
```

Вызов события

9) Синхронизация анимации с частотой монитора – requestAnimationFrame

```
let requestId = requestAnimationFrame(callback)

// отмена запланированного запуска callback
cancelAnimationFrame(requestId);
```

Анонимные функции

1) **Может быть создана там, где не нужно создание отдельного кода**

```
button.addEventListener('click', function (event) { // эта анонимная функция будет вызвана при клике }, false);
```

2) Может быть присвоена переменной

```
let anonym = function func() {   console.log("это анонимная функция"); };   anonym();
```

Стрелочные функции

```
let func = (arg1, arg2, ...argN) => expression

let sum = (a, b) => a + b;

let sum = (a, b) => { // фигурная скобка, открывающая тело многострочной функции
let result = a + b;
return result; // при фигурных скобках для возврата значения нужно явно вызвать return };
```

Задание

Создайте персонаж и окружающий фон.

Задайте перемещение персонажа при помощи клавиш по двум координатным осям – Ох и ОУ