

1. Какие есть способы объявления функций?
 - Function Declaration (основной)
 - Function Expression
 - Стрелочные функции (самый новый)
2. Приведите примеры вызова одной и той же функции всеми известными вам способами.
 - `function sum(a, b) {`
 `return a + b;`
 `}`
 - `let sum = function(a, b) {`
 `return a + b;`
 `};`
 - `const sum = (a, b) => a + b;`
3. В чем разница между тестированием и отладкой (дебаггингом)? А что такое логирование?

Отладка - в процессе разработки ЕЩЕ НЕ ГОТОВОЙ программы. Выявление и устранение ошибок.

Тестирование - получение рабочих характеристик, а также подтверждение работоспособности ГОТОВОЙ программы.

Речь идет о записи специального текстового файла (лога) с полной информацией о работе программы, действиях пользователей. В результате получается некий журнал, каждая строка в котором соответствует определенному действию.

4. В чем разница между Function Expression и Function Declaration?
 - Function Declaration: функция объявляется отдельной конструкцией «function...» в основном потоке кода.
 - Function Expression: функция, созданная внутри другого выражения или синтаксической конструкции. В данном случае функция создается в правой части «выражения присваивания» =:

Более тонкое отличие состоит, в том, когда создается функция движком JavaScript.

Function Expression создается, когда выполнение доходит до него, и затем уже может использоваться.

После того, как поток выполнения достигнет правой части выражения присваивания `let sum = function...` – с этого момента, функция считается созданной и может быть использована (присвоена переменной, вызвана и т.д.).

С Function Declaration всё иначе.

Function Declaration можно использовать во всем скрипте (или блоке кода, если функция объявлена в блоке).

Другими словами, когда движок JavaScript готовится выполнять скрипт или блок кода, прежде всего он ищет в нём Function Declaration и создаёт все такие функции. Можно считать этот процесс «стадией инициализации».

5. Что делает функция `console.log()`?

Если нужно что-то вывести в консоль из кода, применяется функция `console.log`

6. Что такое BOM и DOM?

Объектная модель браузера (Browser Object Model, BOM) – это дополнительные объекты, предоставляемые браузером (окружением), чтобы работать со всем, кроме документа.

7. Есть вот такая страница:

```
<!DOCTYPE HTML>
<html>
  <body>
    <form name="search">
      <label>Поиск:
        <input type="text" name="search"> </label>
      <input type="submit" value="Search!"> </form>
    <hr>
    <form name="search-person">Поиск по посетителям:
      <table id="age-table">
        <tr>
          <td>Возраст:</td>
          <td id="age-list">
            <label>
              <input type="radio" name="age"
value="young">до 18</label>
            <label>
              <input type="radio" name="age"
value="mature">18-50</label>
            <label>
              <input type="radio" name="age"
value="senior">старше 50</label>
          </td>
        </tr>
        <tr>
          <td>Дополнительно:</td>
          <td>
            <input type="text" name="info">
            <input type="text" name="info">
            <input type="text" name="info"> </td>
          </tr>
        </table>
        <input type="button" value="Search!"> </form>
      </body>
    </html>
```

Как найти в ней?...

1. Таблицу с `id="age-table"`
`let elem = document.getElementById('age-table');`
2. Все элементы `label` **внутри** этой таблицы (их три)
`let elements=document.getElementsByTagName('label');`
3. Форму `form` с именем `name="search-person"`
`let elements=document.getElementsByName('search-person')`

8. Как сделать переход на другую страницу при клике на кнопку (без , только средствами JavaScript)?

```
alert(location.href); // показывает текущий URL
if (confirm("Перейти на Wikipedia?")) {
    // перенаправляет браузер на другой адрес
    location.href = "https://wikipedia.org";
}
```

9. Как можно обнулить (очистить) значение внутри input?

- Очистить значение поля ввода, либо любого другого DOM элемента имеющего свойство value, можно просто добавив обработчик к элементу. Недостатком этого подхода, является необходимость добавления обработчика к каждому элементу и помимо этого, элемент должен существовать на момент добавления обработчика:

```
document.getElementById("clearButton").onclick = function(e) {
    // Если необходимо предотвратить/отменить событие по умолчанию,
    // то необходимо вызвать метод preventDefault у события
    // https://developer.mozilla.org/ru/docs/Web/API/Event/preventDefault
    // e.preventDefault();
    // если необходимо также предотвратить дальнейшее "всплытие" события,
    // то необходимо вызвать метод stopPropagation у события
    // https://developer.mozilla.org/ru/docs/Web/API/Event/stopPropagation
    // e.stopPropagation();
    document.getElementById("textInput").value = "";
}
<input id="textInput" type="text" value="abcdefgh" placeholder="Input any string value" />
<button id="clearButton">Clear</button>
```

10. Как будет выглядеть ваша функция приветствия из прошлого домашнего задания, если ее переписать в стрелочном формате?

```
let showMessage=(a = prompt("Ваше Имя?", ))=>alert(`Привет, ${a}!`)
```