Задание для ИСП-420п от (23.12.23) (ПИТПМ)

**1.Основы JavaScript:**

* **Каковы типы данных есть в JavaScript? Составить диаграмму. (Figma, diagrams.net и аналогичные)**

**Одной из основных особенностей Java является то, что данный язык является строго типизированным. А это значит, что каждая переменная и константа представляет определенный тип и данный тип строго определен. Тип данных определяет диапазон значений, которые может хранить переменная или константа.**

**boolean: хранит значение true или false**

**byte: хранит целое число от -128 до 127 и занимает 1 байт**

**short: хранит целое число от -32768 до 32767 и занимает 2 байта**

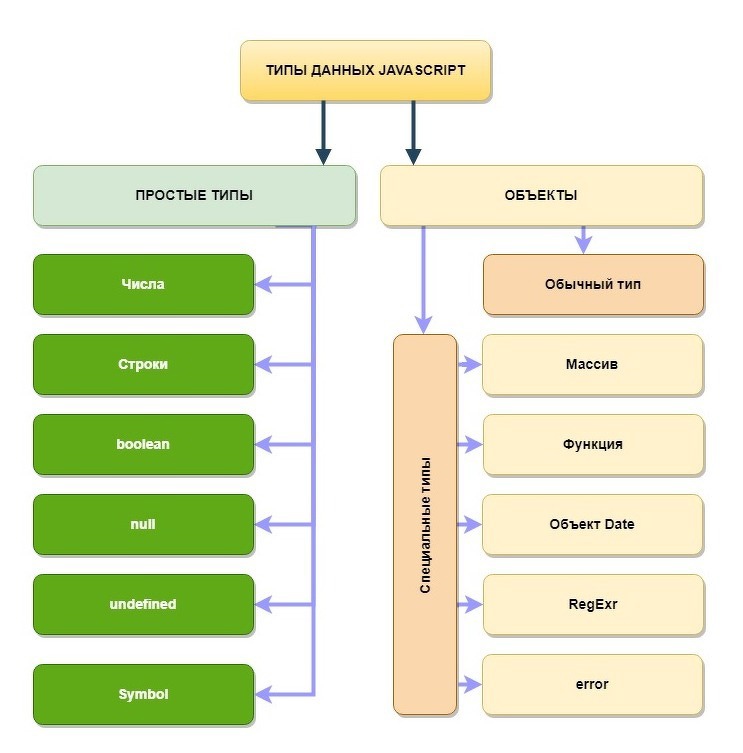
**int: хранит целое число от -2147483648 до 2147483647 и занимает 4 байта**

**long: хранит целое число от –9 223 372 036 854 775 808 до 9 223 372 036 854 775 807 и занимает 8 байт**

**double: хранит число с плавающей точкой от ±4.9\*10-324 до ±1.7976931348623157\*10308 и занимает 8 байт**

**float: хранит число с плавающей точкой от -3.4\*1038 до 3.4\*1038 и занимает 4 байта**

**char: хранит одиночный символ в кодировке UTF-16 и занимает 2 байта, поэтому диапазон хранимых значений от 0 до 65535**

****

* **В чем разница между let, const и var при объявлении переменных?**

**Для создания переменной используется ключевое слово let, const или var**

**Чаще всего, при объявлении переменной ей устанавливают стартовое значение при помощи оператора присваивания =. Тип значения может быть абсолютно любым —**[**строка**](https://doka.guide/js/string/)**,**[**число**](https://doka.guide/js/number/)**,**[**объект**](https://doka.guide/js/object/)**,**[**массив**](https://doka.guide/js/arrays/)**и так далее.**

**Затем переменную можно использовать как заменитель значения в коде:**

**Перед выполнением скрипта JavaScript находит код создания переменных и заранее создаёт их. Получается, что в начале выполнения скрипта все переменные, описанные в коде, уже объявлены.**

**Используя ключевое слово let, можно объявить переменную без присвоения ей начального значения. В таком случае она будет равна undefined**

**При помощи const нельзя объявлять переменные без значения**

**К переменным let и const нельзя обращаться до их объявления в коде**

**Оба типа переменных имеют блочную**[**область видимости**](https://doka.guide/js/closures/)**и не становятся частью глобального объекта (window в браузере, global в Node.js). Блочная область видимости не даёт получить значение переменной вне блока, где она была объявлена.**

**Если объявить переменные внутри блока [if](https://doka.guide/js/if-else/), то обращение к ним вне блока будет выбрасывать ошибку**

**Значение в переменной, созданной через let, можно изменять**

**Стартовое значение const изменить нельзя, будь то примитивное значение или ссылка на объект**

**Однако объект, хранящийся в const, можно мутировать.**[**Объекты хранятся по ссылке**](https://doka.guide/js/ref-type-vs-value-type/#ssylochnye-tipy-dannyh)**, и изменение объекта не приводит к изменению ссылки на него**

**Переменные var**

**Переменные var можно объявлять без присвоения им значения, в таком случае они будут равны undefined**

**Переменные, объявленные через var, имеют функциональную область видимости. Они доступны только в пределах текущей функции или глобального объекта, если функции нет**

**Объявление переменных вне функций делает их глобальными переменными. Они доступны как свойства глобального объекта**

**К переменным, объявленным при помощи ключевого слова var, можно обращаться до момента объявления. В отличие от let и const, ошибки это не вызовет. Такое поведение называется hoisting - «всплытие»**

* Что такое hoisting

hoisting — это механизм в JavaScript, в котором переменные и объявления функций, передвигаются вверх своей области видимости перед тем, как код будет выполнен. Ещё до выполнения кода интерпретатор JavaScript загружает в память функции, объявленные как. function declaration, а также объявления переменных и функций function expression, но не их значения.

**2. Функции**

* **Как объявить функцию в JavaScript?**

**Функция объявляется с помощью ключевого слова function.**

**Основные правила именования функции аналогичны именованию переменной. Лучше написать описательное имя для функции. Например, если функция используется для сложения двух чисел, можно назвать функцию add или addNumbers.**

**Тело функции записывается внутри {}.**

* В чем разница между function declaration и function expression?

Function Declaration – функция, объявленная в основном потоке кода.

Function Expression – объявление функции в контексте какого-либо выражения, например присваивания.

* Что такое стрелочные функции?

Это — анонимные функции с особым синтаксисом, которые принимают фиксированное число аргументов и работают в контексте включающей их области видимости, то есть — в контексте функции или другого кода, в котором они объявлены.

**3. Объекты и**[массивы](http://m.vksit.ru/mod/page/view.php?id=8616)**:**

* Как создать объект в JavaScript?

Как и все переменные JavaScript, имя объекта (которое тоже может быть переменной) и имя свойства являются чувствительными к регистру. Вы можете определить свойство указав его значение.

* Как добавить новый элемент в конец массива?

Чтобы добавить элемент в конец массива, используют метод push ().

* Как перебрать все элементы массива?

Специальная версия цикла for предназначена для перебора элементов в наборах элементов

**4. Обработка событий**

* Как назначить обработчик [события](http://m.vksit.ru/mod/page/view.php?id=6116) элементу?

через HTML-атрибут on{событие} (не является хорошей практикой);

посредством свойства DOM-элемента on{событие};

используя специальный метод addEventListener.

* Что такое event delegation?

event delegation это шаблон, основанный на концепции Пузырькового отображения событий. Это шаблон обработки событий, который позволяет вам обрабатывать события на более высоком уровне в дереве DOM, отличном от уровня, на котором событие было получено впервые.

* Как отменить стандартное поведение [события](http://m.vksit.ru/mod/page/view.php?id=6116)?

preventDefault() — метод [события](https://doka.guide/js/event/). Этот метод отменяет поведение браузера по умолчанию, которое происходит при обработке события.

**5. Асинхронность**

* Как работает синхронный и асинхронный код в JavaScript?

Выполнение JS-кода — однопоточное. Это значит, что в конкретный момент времени движок может выполнять *не более одной* строки кода. То есть вторая строка не будет выполнена, пока не выполнится первая.

Такое выполнение кода (строка за строкой) называется синхронным.

* Что такое Promise?

Промис (Promise) — специальный объект JavaScript, который используется для написания и обработки асинхронного кода.

Асинхронные функции возвращают объект Promise в качестве значения. Внутри промиса хранится результат вычисления, которое может быть уже выполнено или выполнится в будущем.

Промис может находиться в одном из трёх состояний:

pending — стартовое состояние, операция стартовала;

fulfilled — получен результат;

rejected — ошибка.

* Как использовать async/await?

Добавленное перед определением функции ключевое слово async делает функцию асинхронной. Возвращаемое значение такой функции автоматически оборачивается в [Promise](https://doka.guide/js/promise/)

Асинхронные функции нужны для выполнения асинхронных операций: работой с API, базами данных, чтения файлов и т.д.

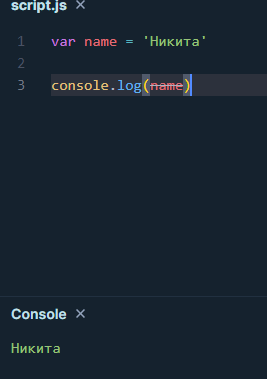
Асинхронные операции выполняются не сразу: код отправил запрос к API и ждёт, пока сервер пришлёт ответ. Ключевое слово await используется, чтобы дождаться выполнения асинхронной операции

Движок JavaScript при этом не блокируется и может выполнять другой код. Как только ответ получен, выполнение кода продолжается

### Задания

**Основы JavaScript:**

* Создайте переменную, содержащую ваше имя, и выведите ее в консоль.

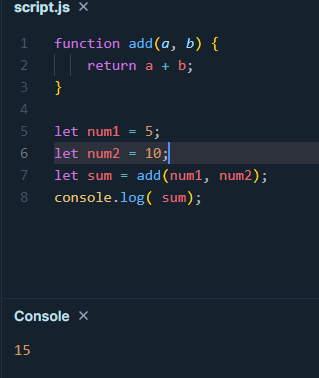


* . Создайте массив чисел и напишите цикл для вывода каждого элемента.

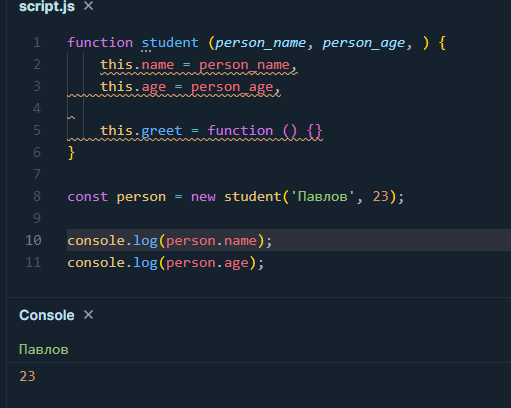


**Функции:**

* Напишите функцию, которая принимает два числа и возвращает их сумму.

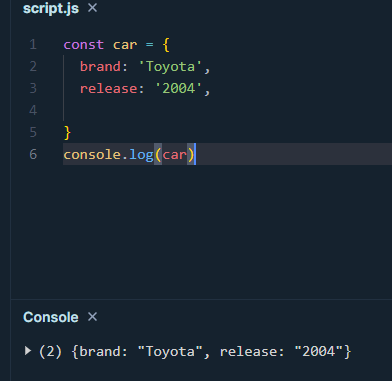


* Создайте функцию-конструктор для объекта "Студент" с полями "имя" и "возраст".

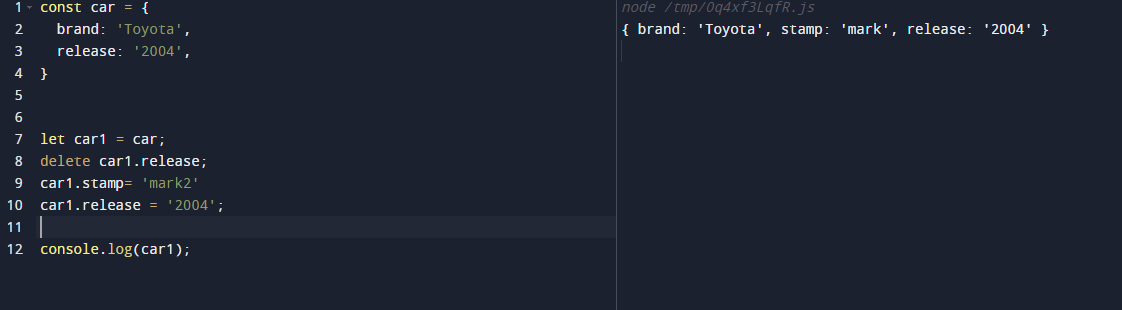


**Объекты и**[массивы](http://m.vksit.ru/mod/page/view.php?id=8616)**:**

* Создайте объект "автомобиль" с полями "марка" и "год выпуска".



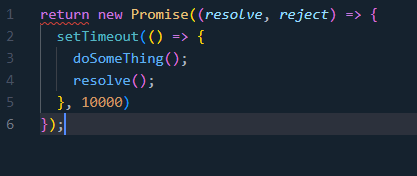
* Добавьте новый элемент в середину массива и объясните, как это сделать.



Копируем объект, называем его car1. В новом объекте удаляем год выпуска, добавляем новый элемент и добавляем год выпуска.

Асинхронность:

* Напишите функцию, возвращающую Promise, который разрешается через 1 секунду.



* Используя async/await, напишите код, который вызывает эту функцию и выводит результат в консоль.

