

Сначала прочитайте памятку о домашнем задании: <https://docs.google.com/document/d/1-wLPp7khnRijyFkAKA4oN2ZGkCxuDkaTwVuXhqBUeo/edit?usp=sharing>

**1.** (это задание делайте по желанию) Написать функцию, преобразующую число в объект. Передавая на вход число в диапазоне [0, 999],

мы должны получить на выходе объект, в котором в соответствующих свойствах описаны разряды числа:

- единицы (в свойстве units)
- десятки (в свойстве tens)
- сотни (в свойстве hundreds)

Например, для числа 45 мы должны получить следующий объект:

```
{  
  units: 5, //это единицы  
  tens: 4, //это десятки  
  hundreds: 0, //это сотни  
}
```

Если число было передано вне [0, 999] диапазона, не целое число или вообще не число, необходимо выдать соответствующее сообщение с помощью console.log и вернуть пустой объект.

*Вам может пригодиться:*

- Number.isInteger(value) – функция проверяет, является ли переданное число целым, подробнее здесь <https://mzl.la/2XCVSsx>
- Math.floor(value) - метод возвращает целое число, которое меньше или равно данному числу (проще говоря округляет в меньшую сторону), подробнее здесь <https://mzl.la/2Qx42SO> .
- Используйте также остаток от деления на 10, например 123 % 10 будет 3
- Вам также может пригодится делить число на 100 и на 10.

**1.1 (это обязательное задание)** Сделайте в стиле es5, а затем в стиле es6 (по аналогии из дополнительных видео -> 3 примеры наследования -> типы на es5 и es6), конструктор Product, который принимает параметры name и price, сохраните их как свойства объекта. Также объекты типа Product должны иметь метод make25PercentDiscount, который будет уменьшать цену в объекте на 25%. Имейте в виду, что метод make25PercentDiscount не должен быть внутри функции-конструктора, и также не нужно создавать отдельный объект-прототип (как объект transport в уроке).

**1.2 (это обязательное задание)** Сделайте в стиле es5, а затем в стиле es6 (по аналогии из дополнительных видео -> 3 примеры наследования -> механика наследования),  
а) конструктор Post, который принимает параметры author, text, date и сохраняет их как свойства объекта. Объекты типа Post должны иметь метод edit, который будет принимать текст и записывать его в свойство text объекта.  
б) конструктор AttachedPost, который принимает параметры author, text, date. Проинициализируйте эти свойства с помощью конструктора Post, чтобы не дублировать код. Также в конструкторе AttachedPost должно создаваться свойство highlighted со значением false. Унаследуйте в объектах типа AttachedPost методы из Post. Объекты типа AttachedPost должны иметь метод makeTextHighlighted, который будет назначать свойству highlighted значение true.

**2** (это задание не является частью учебной программы, делайте его по желанию). Для игры бродилка (которая есть в дополнительных видео), добавить возможность ходить по диагонали цифрами 1, 3, 7, 9. Также необходимо сделать так, чтобы пользователь не мог совершить шаг в стенку, т.е. при направлении в стенку и игрок оставался на том же месте где стоял.

**3** (это задание не является частью учебной программы, делайте его по желанию). На базе игры (приняв за пример), созданной в дополнительных видео, реализовать игру «Кто хочет стать миллионером?». Т.е. у вас должен быть главный объект, содержащий всю логику игры, который будет иметь методы, например метод `run`, возможно метод `init` и т.д.

В игре должны быть заранее подготовлены список вопросов и ответов (как минимум 5 вопросов).

Игра должна приветствовать пользователя, после чего задавать вопросы пользователю и предлагать варианты ответов в виде теста, например:

Сколько букв в слове "привет":

- a. Пять.
- b. Шесть.
- c. Семь.
- d. Куда я попал?

Проверять правильный вариант выбрал пользователь или нет, необходимо вести счет.

По окончании игры, когда было задано 5 вопросов, вы должны сообщить пользователю его счет и предложить сыграть снова.

Также должна быть возможность выхода из игры заранее, если пользователю надоело играть.