МШП 20.35 УНИВЕРСИТЕТ КОД БУДУЩЕГО Минцифры России НАЦИОНАЛЬНЫЕ ПРОЕКТЫ РОССИИ ЦИФРОВАЯ ЭКОНОМИКА

Fullstack-разработка

JavaScript
Массивы
Объекты

МШП

Цели урока:

1. Разберёмся, как правильно хранить разные данные
2. Узнаем новый тип данных

МШП

Разбор ДЗ

Что получилось?

Что было трудно?

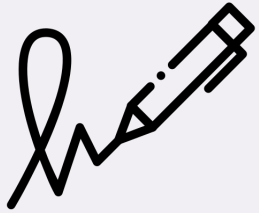
МШП

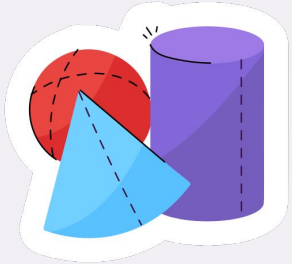
Теория

Структуры данных

МШП

Любой объект в мире несёт в себе какую-то важную информацию.

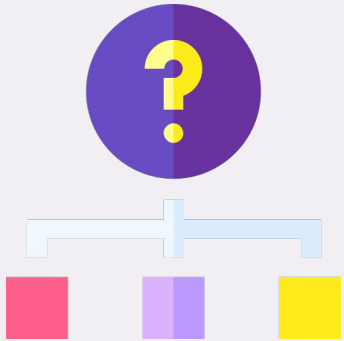





Когда мы разрабатываем приложения, то отвечаем себе на два вопроса:

1. **Какую информацию мне нужно хранить в коде?**

2. **Как её лучше хранить?**



МШП

Ситуация 1

В игре ночь, у меня 100 монет,
передо мной противник Жадный гном...

```
let player = `Жадный гном`;
let price = 100;
let isDark = true;
```

Если данные **разрозненные,**
то их удобно хранить в **разных**
переменных.

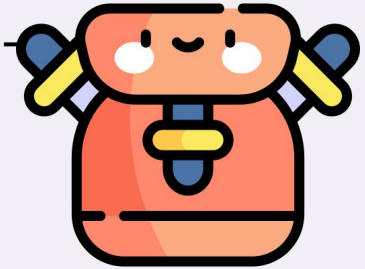
МШП

Ситуация 2

В рюкзаке персонажа лежат предметы.

За каждым закреплена своя горячая клавиша.

```
let backpack = [
    `Меч`, `Посох`, `Зелье`, `Руна`
];
```



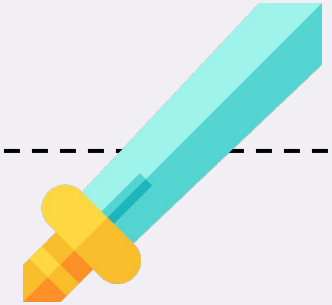
Если данные однотипные
и их порядок важен, то
их нужно хранить в **массиве**

МШП

Ситуация 3

Игроку нужно рассказать про все особенности меча, который он планирует купить.

```
название - Горный хрусталь
атака - 50 единиц урона
цена - 100 монет
тип - одноручный
```



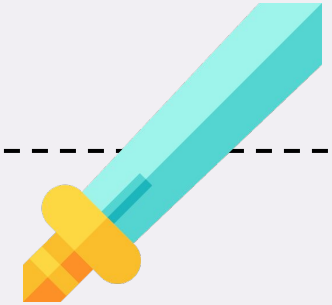
Если нужно хранить данные про **один объект,** то как его лучше хранить?

МШП

Ситуация 3

Игроку нужно рассказать про все особенности меча, который он планирует купить.

```
let name = `Горный хрусталь`;
let attack = 50;
let price = 100;
let single = true;
```



Для одного меча можно сделать переменные, но что, если в продаже 100 мечей?

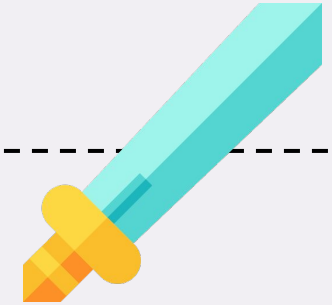
МШП

Ситуация 3

Мы уже делали так раньше в задачах.

Удобно?

```
let names = [ ... ];
let attacks = [ ... ];
let prices = [ ... ];
let singles = [ ... ];
```



НЕТ

МШП

Объект

```
let sword = {
    name: `Горный хрусталь`,
    attack: 50,
    price: 100,
    single: true
};
```

Объект – это структура данных, которая объединяет в себе разные **свойства** и их **значения**

МШП

Чтение свойства

```
let sword = {
    name: `Горный хрусталь`,
    attack: 50,
    price: 100,
    single: true
};

console.log( sword.name );
```

Через **имя свойства**

можно **прочитать** его **значение**

МШП

Изменение свойства

```
let sword = {
    name: `Горный хрусталь`,
    attack: 50,
    price: 100,
    single: true
};

sword.price = 150;
```

Через **имя свойства**

можно **перезаписать** его **значение**

МШП

Другие объекты

- `sword.price = 150;`

Ничего не напоминает?

- `node.innerHTML`
- `input.value`
- `string.length`

Это тоже объекты

МШП

Объекты очень важны
в языке Javascript!

Настолько, что объект считают
отдельным типом данных.

А какие типы данных вы ещё
знаете?

МШП

Практика

Бард

МШП

Теория

Ключи объекта

МШП

Объект очень похож на волшебную сумку Гермионы Грейнджер

1. Можно разместить сколько угодно предметов

2. Можно достать нужный предмет, если знать его название

МШП

Ключи объекта

Когда мы создаём **объект**, то придумываем названия его **свойствам**.

Имя свойства называют **ключом.**

```
let bag = {
    book: `Зельеварение`,
    snake: 2
};
```

МШП

Ключи объекта

Ключ можно записать двумя способами:

в кавычках и **без кавычек**.

```
let bag = {
    snake: 2
};
```

```
let bag = {
    "snake": 2
};
```

Благодаря этому ключ может быть **любым:**

```
let bag = {
    "змея!!": 2
};
```

Важно!
Кавычки
не косые

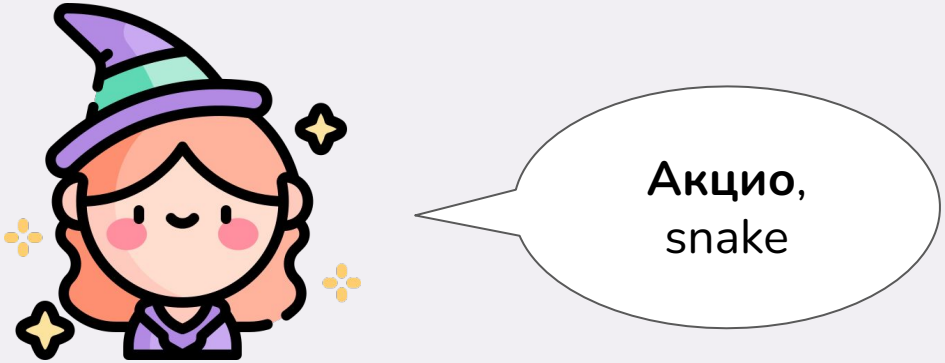
МШП

Ключи объекта

Также существуют два способа прочитать значение по ключу:

```
bag.snake
```

```
bag[`snake`]
```



МШП

Ключи объекта

Благодаря этому можно магически доставать из объекта то, что попросит пользователь:

```
let key = input.value;

console.log( bag[key] );
```

МШП

Ключи объекта

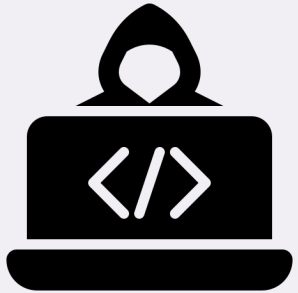
А если не найдётся то, что мы искали?
Лучше проверить!

```
let key = input.value;
let item = bag[key];
if (item) {   /* нашли */  }
```

Акцио,
homework

МШП

Давайте вместе разберём этот приём на задаче!



МШП

Практика

Сумка Гермионы

Переводчик

МШП

Теория

Объекты и массивы

МШП

Объект

В объект можно положить **что угодно!**

Любой тип данных:

```
let sword = {
    name: `Горный хрусталь`,
    attack: 50,
    price: 100,
    single: true
};
```

МШП

Объект

В объект можно положить **что угодно!**

Массив:

```
let user = {
    name: `Михаил`,
    friends: [`Коля`, `Настя`]
};
```

МШП

Объект

В объект можно положить **что угодно!**

Другой объект:

```
let user = {
    name: `Михаил`,
    friends: [`Коля`, `Настя`],
    secure: {
        phone: `79999999999`,
        email: `test@mail.ru`
    },
};
```

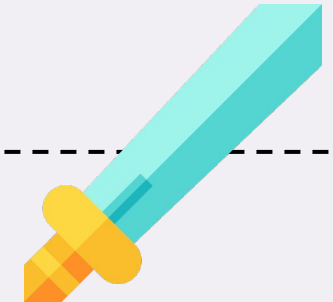
МШП

Ситуация 3

Вернёмся к ситуации из начала урока.

Как можно хранить информацию о нескольких мечах более удобно?

```
let names = [ ... ];
let attacks = [ ... ];
let prices = [ ... ];
let singles = [ ... ];
```



МШП

Массив объектов

Правильно!

Создать массив из объектов!

```
let array = [
    { ... }, // меч 1
    { ... }, // меч 2
    { ... }, // меч 3
    { ... }, // меч 4
    { ... }  // меч 5
];
```

МШП

Время испробовать этот приём на практике.

Удачи!

МШП

Практика

Покемоны

МШП

Итоги

- Для хранения однотипных данных используют **массивы**
- Для хранения связанных по смыслу данных используют **объекты**
- **Можно создавать объект с массивами и массив с объектами**

МШП

Цели урока:

- ✓ Разобрались, как правильно хранить разные данные
- ✓ Узнали новый тип данных

МШП

Домашнее задание