12. Типы данных

1. Что будет выведено на экран?

console.log('hex' \* 'let');

Начало формы



NaN



'hexlet'



На экран ничего не выведется



9

Конец формы

2. Что будет выведено на экран?

console.log(typeof '1234');

Начало формы



int



NaN



string

Конец формы

3. Всегда ли при выводе на экран с помощью console.log() можно однозначно сказать, с каким типом данных мы имеем дело?

Начало формы



Нет, так как console.log() в конечном счете выводит на экран текстовую строку



Да. Перед выводимым значением указывается его тип

Конец формы

1. NaN

2. string

3.   
Нет, так как console.log() в конечном счете выводит на экран текстовую строку

13. Неизменяемость и примитивные типы

1. Какое значение содержится в переменной num?

|  |  |
| --- | --- |
|  | let num = -55; |
|  | console.log(-num); |

Начало формы



55



None



-55

Конец формы

2. Можно ли перезаписывать значение переменных?

Начало формы



Значение переменной невозможно изменить



Нет, изменение значения переменных недопустимо



Можно, но делать это стоит только по необходимости (например, в циклах)

Конец формы

3. Можно ли изменить строку в JavaScript?

Начало формы



Да, строка – это изменяемый тип данных



Нет, строка является примитивным типом данных



Да, все типы данных в JavaScript изменяемы

Конец формы

1. -55

2. Можно, но делать это стоит только по необходимости (например, в циклах)

3. Нет, строка является примитивным типом данных

14. Функции

1. Какие операции могут выполнять функции?

Начало формы



Функции могут выполнять любые допустимые в программировании операции



Только математические вычисления

2. Нужно ли знать наизусть все названия функций?

Начало формы



Нет. Существуют подсказки редактора и документация по функциям



Обязательно. Иначе невозможно будет их использовать

Конец формы

3. Дан следующий код:

|  |  |
| --- | --- |
|  | const getOne = () => { |
|  | return 1; |
|  | } |
|  |  |
|  | console.log(getOne); |

В этом случае вызов функции getOne() возвращает единицу. Выберите верное утверждение относительно этого кода:

Начало формы



На экран будет выведено число 1 — результат работы функции



код не выведет число 1 на экран, потому что функция getOne() не вызывается



Код ничего не выведет на экран, потому что функция getOne() только возвращает значение

Конец формы

Конец формы

1. Функции могут выполнять любые допустимые в программировании операции

2. Нет. Существуют подсказки редактора и документация по функциям

3. код не выведет число 1 на экран, потому что функция getOne() не вызывается

### 15. [Сигнатура функции](https://ru.hexlet.io/courses/js-basics/lessons/signature/theory_unit)

1. Что произойдет, если вызвать стандартную функцию Math.sqrt(), не передавая в нее никаких параметров?

Начало формы



Функция выполнится и вычислит квадратный корень нужного вам числа



Функция выполнится и вернет NaN



Функция выполнится и вычислит квадратный корень случайного числа

Конец формы

2. Можно ли использовать значения параметров по умолчанию?

Начало формы



Да, если такие параметры определены в функции



Да, все параметры являются необязательными



Нет, функции всегда должны принимать одинаковое количество параметров

Конец формы

3. Существуют ли функции, которые возвращают сразу 10 значений?

Начало формы



Нет, любая функция может вернуть только одно значение



Функции всегда возвращают случайное количество значений



Да, количество возвращаемых значений ограничено только объемом оперативной памяти

Конец формы

1. Функция выполнится и вернет NaN

2. Да, если такие параметры определены в функции

3. Нет, любая функция может вернуть только одно значение

16. Вызов функции

1. Функция getOne() возвращает единицу. Что будет выведено на экран?

console.log(getOne());

Начало формы



Вызов функции – выражение. На экран будет выведен результат ее работы – 1 (единица)



Ничего, так как функция getOne() сама ничего не выводит на экран, она только возвращает значение



Будет выведено тело функции

Конец формы

2. Выберите верное утверждение:

Начало формы



Если слишком увлекаться промежуточными переменными, то память компьютера расходуется неэффективно



Разделение вызовов на промежуточные вычисления облегчает чтение кода и его отладку



Лучше не разделять вызовы на промежуточные вычисления, так код получается короче

Конец формы

3. Функция sum() принимает два числа и возвращает их сумму. Взгляните на код ниже. Какое число будет выведено на экран?

|  |  |
| --- | --- |
|  | let result = sum(sum(1, 3), sum(sum(4, 2), 3)); |
|  | console.log(result); // => ? |

Начало формы



14



12



13

Конец формы

16. 1. Вызов функции – выражение. На экран будет выведен результат ее работы – 1 (единица)

2. Разделение вызовов на промежуточные вычисления облегчает чтение кода и его отладку

3. 13

17.

1. Как в сигнатуре функции обозначают необязательность параметров?

Начало формы



{, ...}



[]



[, ...]

2. Выберите верные утверждения:

*(нужно выбрать все корректные ответы)*

Начало формы



функция может принимать переменное количество параметров



параметры функции могут быть необязательными



количество параметров функции всегда задается жестко

3. Может ли функция принимать переменное число параметров?

Начало формы



нет, не может



да, может

Конец формы

Конец формы

1. []

2. функция может принимать переменное количество параметров



параметры функции могут быть необязательными

3. да, может

18. Детерменированность

1. Является ли функция length() из урока детерминированной?

Начало формы



Нет, эта функция изменяет исходное число



Нет, мы не знаем в каком виде будет возвращено число



Да, на одних и тех же данных функция всегда вернет одинаковые значения

Конец формы

2. Выберите верные утверждения

*(нужно выбрать все корректные ответы)*

Начало формы



Печать на экран не является побочным эффектом



Функции с побочными эффектами нужно использовать осторожно, они могут стать источником проблем в коде



Операция печати на экран – это аналог возврата значения из функции



Печать на экран является побочным эффектом функции

Конец формы

3. Какое значение будет выведено на экран?

Math.random();

Начало формы



Невозможно предсказать, т.к. функция Math.random() недетерминированная



0.0985661311319767



1.0985661311319767

Конец формы

1. Да, на одних и тех же данных функция всегда вернет одинаковые значения

2. Печать на экран является побочным эффектом функции

Функции с побочными эффектами нужно использовать осторожно, они могут стать источником проблем в коде

3. Невозможно предсказать, т.к. функция Math.random() недетерминированная

19. 1. Что такое стандартная библиотека?

Начало формы



В JavaScript нет стандартных функций, все функции нужно писать самому



Набор функций, которые поставляются вместе с JavaScript



Набор функций, которые можно использовать в своем коде на JavaScript

Конец формы

2. Можно ли узнать, как работает функция из стандартной библиотеки?

Начало формы



Нет, такие функции можно использовать только вслепую



Написать разработчику



Изучить сигнатуру и способ использования в документации

Конец формы

3. Как узнавать о существующих методах в стандартной библиотеке?

*(нужно выбрать все корректные ответы)*

Начало формы



В JavaScript достаточно знать несколько функций, чтобы писать программы любой сложности



Периодически нужно изучать документацию по стандартным функциям



Читать чужой код на гитхаб

Конец формы

1.   


Набор функций, которые поставляются вместе с JavaScript

2. Изучить сигнатуру и способ использования в документации

3.

Периодически нужно изучать документацию по стандартным функциям



Читать чужой код на гитхаб

20.

1. Что будет выведено на экран?

|  |  |
| --- | --- |
|  | let company = 'hexlet'; |
|  | console.log(company.toUpperCase()); // => ? |

Начало формы



hexlet



HEXLET



У строки нет такого метода



Hexlet

Конец формы

2. Выберите верные утверждения

*(нужно выбрать все корректные ответы)*

Начало формы



У данных нет методов, есть только свойства



Методы в JavaScript – это обычные функции



Метод применяется к данным, на которых он вызван



Метод – это функции, находящиеся внутри свойств объекта

Конец формы

3. Какое значение содержится в переменной company?

|  |  |
| --- | --- |
|  | let company = 'Hexlet'; |
|  | console.log(company.toLowerCase().toUpperCase()); |

Начало формы



hexlet



Код завершится с ошибкой



Hexlet



HEXLET

Конец формы

1. HEXLET

2. Метод – это функции, находящиеся внутри свойств объекта



Метод применяется к данным, на которых он вызван

3. Hexlet

Конец формы

Конец формы