26. 1. Когда можно не писать слово return для возврата результата из функции?

Начало формы



Никогда



Всегда, когда функцию можно разместить на одной строке



Когда используется сокращенный синтаксис

Конец формы

2. Какое из определений функции верно?

Начало формы



const double = (x) => { x \*\* 2; }



const double = (x) => x \*\* 2;



const double = (x) => return x \*\* 2;

Конец формы

3, Во всех ли версиях языка доступен сокращенный синтаксис определения функции?

Начало формы



Да, в том числе и в самых старых



Только в самой последней



Начиная с версии es6

Конец формы

27. 1. Что будет выведено на экран?

console.log(5 <= 5);

Начало формы



false



5



true

2. Как правильно назвать функцию, которая проверяет, есть ли у пользователя друзья, и возвращает результат булева типа?

Начало формы



hasFriends()



isFriends()



getFriends()

3. Как называются функции, которые отвечают на вопрос «да» или «нет» и возвращают значение типа boolean?

Начало формы



Такие функции не имеют отдельного названия



Функции-предикаты



Функции-прейскуранты



Утвердительные функции

Конец формы

28.1. Каким будет результат (12 > 40) || (400 >= 400)?

Начало формы



Невозможно ответить



false



true

Конец формы

2. Как работает оператор отрицания !?

Начало формы



Меняет логическое значение на противоположное

Всегда возвращает false

Отменяет действие других операторов

Конец формы

3. Как в коде обозначается оператор И?

Начало формы

||

or

В Javascript нет такого оператора

and

&&

Конец формы

4. Что будет выведено на экран? :)

|  |  |
| --- | --- |
|  | let value = true; |
|  | console.log(!!!!value); |

Начало формы

false

'Заберите меня отсюда!!!!'

true

5. Что будет записано в переменную result?

let result = !true || false

Начало формы

true

На этот вопрос невозможно ответить

false

Конец формы

Конец формы

29. 1. Что будет выведено на экран?

console.log(true || 'yes');

Начало формы

false

*yes*

true

Конец формы

2. Что будет выведено на экран?

console.log(false || 'yes');

Начало формы

*yes*

false

true

Конец формы

3. Какие из значений преобразуются в false?

*(нужно выбрать все корректные ответы)*

Начало формы

'0'

null

NaN

''

undefined

0

Конец формы

30. 1. Дан код:

|  |  |
| --- | --- |
|  | if (a > 100) { |
|  | something; |
|  | } else if (a === 95) { |
|  | somethingDifferent; |
|  | } else if (a === 90) { |
|  | somethingDifferentYet; |
|  | } else { |
|  | somethingElse; |
|  | } |

Что будет, если ни одно условие из блоков if и else if не будет истинным?

Начало формы

будет выполнена инструкция somethingDifferentYet

компьютер зависнет

будет выполнена инструкция somethingElse

Конец формы

2. Дан код:

|  |  |
| --- | --- |
|  | if (temperature > 10) { |
|  | return temperature; |
|  | } else if (temperature < 10) { |
|  | return 0; |
|  | } |

Изменится ли функциональность, если поменять местами условия таким образом:

|  |  |
| --- | --- |
|  | if (temperature < 10) { |
|  | return 0; |
|  | } else if (temperature > 10) { |
|  | return temperature; |
|  | } |

Начало формы

Нет

Да

Конец формы

3. Дан код:

|  |  |
| --- | --- |
|  | if (temperature > 10) { |
|  | return temperature; |
|  | } else if (temperature < 10) { |
|  | return 0; |
|  | } |

Изменится ли функциональность, если изменить else if на else?

Начало формы



Нет



Да

Конец формы31. 1. Проанализируйте следующий код:

|  |  |
| --- | --- |
|  | // функция-предикат определяет, является ли число num четным |
|  | const isEven = (num) => { |
|  | return num % 2 === 0; |
|  | } |
|  |  |
|  | // функция увеличивает полученное число num на 10 |
|  | const increaseNum = (num) => { |
|  | return num + 10; |
|  | } |
|  |  |
|  | const num = 6; |
|  |  |
|  | const result = isEven(increaseNum(num) + 1) ? num + 1 : 1 - num; |

Что будет находиться в переменной result после его выполнения?

Начало формы

true

-5

16

7

17

Конец формы

Конец формы

2. Что можно сказать о коде ниже?

|  |  |
| --- | --- |
|  | const num = 3; |
|  | const result = num === 2 ? 'two' : num === 3 ? 'three' : 'other number'; |

Начало формы



Ничего особенного, код как код



В коде один тернарный оператор вложен в другой. Так делать категорически нельзя. Кстати, в result будет записана строка *three*



В результате в переменной result окажется строка *two*

Конец формы

3. Этот скрипт определяет оценку в зависимости от набранных в тесте баллов:

|  |  |
| --- | --- |
|  | const testScore = 10; |
|  | const result = testScore < 10 ? 'bad' : 'good'; |
|  | console.log(result); |

Что будет выведено на экран?Начало формы



goodbadundefined

32. 1.

|  |  |
| --- | --- |
|  | if (answer === 10 || answer === 11) { |
|  | return "Good!"; |
|  | } else if (answer === 12) { |
|  | return "Better!"; |
|  | } else { |
|  | return "Bad..."; |
|  | } |

Можно ли подобное условие переписать с помощью switch?

Начало формы

Да

Нет

Конец формы

2. Что будет, если при использовании switch внутри case нет ни break, ни return?

Начало формы

произойдет выход из switch

управление будет передано следующему кейсу, если он есть, либо продолжит выполняться код после switch

произойдет выход из функции

управление сразу будет передано в default кейс

Конец формы

3. Выберите главный случай, когда стоит использовать switch.

Начало формы

Эту конструкцию нужно использовать всегда. Чем чаще тем лучше!

Когда в коде есть бесконечный цикл

Если есть цепочка *if else* с проверками на равенство

Конец формы

33. 1. Возможно ли, чтобы код внутри цикла while ни разу не выполнился?

Начало формы

Возможно. Проверка условия выполняется до выполнения, и если условие ложное, то код из тела цикла выполняться не будет

Нет, невозможно. Код в цикле выполняется как минимум один раз

Конец формы2.

Возможны ли бесконечные циклы?

Начало формы

Нет, условие всегда меняется, так что в какой-то момент оно обязательно станет ложным

Да, если условие всегда остается истинным

Конец формы

3. Чему должно быть равно a, чтобы в результате получилось 0?

Начало формы

let a = ;

a -= 8 - a;

Конец формы4. Что будет выведено на экран?

|  |  |
| --- | --- |
|  | let a = 5; |
|  | a -= 4; |
|  | a += 1; |
|  | console.log(a); |

Начало формы

4

2

1

5

Конец формы

34. 1. Какой нейтральный элемент у операции умножения?

Начало формы

1

0

Эта операция не имеет нейтрального элемента

"" (пустая строка)

Конец формы

2. Функция multiplyNumbersFromRange() перемножает числа указанного диапазона. Дополните недостающие места в коде:

Начало формы

const multiplyNumbersFromRange = (start, finish) => {

let i = start;

let multiply = ;

while (i <= finish) {

multiply \*= ;

i += 1

}

return multiply;

}

multiplyNumbersFromRange(3, 5); // 60

3. Какие типы данных можно собирать (агрегировать) в цикле?

*(нужно выбрать все корректные ответы)*

Начало формы



строки



числа



цикл не предназначен для агрегации данных

Конец формы35. 1. Что является нейтральным элементом для строк?

Начало формы



Эта операция не имеет нейтрального элемента



0



1



"" (пустая строка)

Конец формы

2, Какое значение содержится в переменной result?

|  |  |
| --- | --- |
|  | const name = 'hexlet'; |
|  | let result = ''; |
|  | let i = 0; |
|  | while (i < name.length) { |
|  | result = `${result}${name[i].toUpperCase()}` |
|  | i += 1; |
|  | } |

Начало формы



"" (пустая строка)



hexlet



HEXLET

3. Выберите верное утверждение:

Начало формы



для формирования строк в цикле используется только интерполяция



циклы подходят для работы как с числами, так и со строками



циклы можно использовать только для обхода строк

36. 1. Можно ли использовать условные конструкции в теле цикла?

Начало формы



Нет. Все необходимые условия нужно указывать в блоке инициализации цикла



В теле цикла можно писать любые инструкции, в том числе и конструкции с условиями

2. Можно ли прервать работу цикла?

Начало формы



Нет, цикл всегда отрабатывает до конца и выполняет все итерации



Можно вернуть значение по условию и остановить цикл

3. Выберите верное утверждение:

Начало формы



С помощью условий в цикле можно оптимизировать работу программы



В каждом цикле должна быть хотя бы одна условная конструкция для выхода из цикла



В строго типизированных языках нет возможности добавлять условные конструкции в тело цикла

4. С помощью чего чаще всего описываются условия в цикле?

Начало формы



Для их описания используется специальный синтаксис edgecase



С помощью отдельных функций, которые используются внутри реализуемой



С помощью if

37.1 Выберите две формы, которые приведут к одинаковому значению переменной x:

*(нужно выбрать все корректные ответы)*

Начало формы

x++

x -= 0

x -= 1

x--

2. Выберите верные утверждения об инкременте и декременте:

*(нужно выбрать все корректные ответы)*

Начало формы



их можно использовать без ограничений



эти операции имеют побочные эффекты



их нужно использовать с осторожностью



в JavaScript код с этими операциями не пройдет проверку линтера

3. Какое значение будет выведено на экран?

|  |  |
| --- | --- |
|  | let x = 5; |
|  | x++; |
|  | console.log(x++); |

Начало формы

5

6

7

4. Какие значения будут выведены на экран?

|  |  |
| --- | --- |
|  | let x = 10; |
|  | while(--x > 5) { |
|  | console.log(x); |
|  | }; |

Начало формы



10, 9, 8, 7, 6



9, 8, 7, 6, 5



9, 8, 7, 6



10, 9, 8, 7, 6, 5

5. Какое слово будет в переменной result в результате работы цикла?

|  |  |
| --- | --- |
|  | const str = 'HELLO'; |
|  | let result = ''; |
|  | for (let i = 0; i < str.length; i++) { |
|  | const current = str[i]; |
|  | ++i; |
|  | result = `${result}${current.toLowerCase()}` |
|  | } |

Начало формы



hlo



ello



hell



llo

38. 1. Дан код, в котором используется цикл for. Можно ли его переписать, используя цикл while?

Начало формы



да



нет

2. Выберите верные утверждения

*(нужно выбрать все корректные ответы)*

Начало формы

в цикле for обязательно должны присутствовать все три элемента: счетчик, условие, изменение счетчика

цикл while не может быть бесконечным

цикл while может быть бесконечным

цикл for может быть бесконечным

3. Функция getHalfString() принимает строку и возвращает половину строки (если длина строки нечетная, то возвращается большая часть). Добавьте недостающие части кода:

Начало формы

const getHalfString = (str) => {

let result = '';

for (let i = 0; i < .length / 2; i += ) {

result += ;

}

return result;

};

getHalfString("12345"); // 123

Конец формы

Конец формы

Конец формы

Конец формы

Конец формы

Конец формы

Конец формы

Конец формы

Конец формы

Конец формы

Конец формы

Конец формы

Конец формы

Конец формы

Конец формы

Конец формы

Конец формы

27

26.1. Когда используется сокращенный синтаксис

2. const double = (x) => x \*\* 2;

3. Начиная с версии es6

27.1. true 2. hasFriends()3. Функции-предикаты

28.1. true 2. Меняет логическое значение на противоположное 3. && 4. True 5. False

29.1. true 2. *Yes 3. null*

*NaN*

*''*

*undefined*

*0*

30. 1. будет выполнена инструкция somethingElse 2. Нет 3. Да

31. 1. -5. 2. В коде один тернарный оператор вложен в другой. Так делать категорически нельзя. Кстати, в result будет записана строка *three 3.* Good

32. `1. Да 2. управление будет передано следующему кейсу, если он есть, либо продолжит выполняться код после switch 3. Если есть цепочка *if else* с проверками на равенство

33. 1. Возможно. Проверка условия выполняется до выполнения, и если условие ложное, то код из тела цикла выполняться не будет. 2. Да, если условие всегда остается истинным

3. 4 . 4. 2

34. 1. 1 2. const multiplyNumbersFromRange = (start, finish) => {

let i = start;

let multiply = ;

while (i <= finish) {

multiply \*= ;

i += 1

}

return multiply;

}

multiplyNumbersFromRange(3, 5); // 60

ответ{{{{{1]]]] 2]]] 3. Числа, строки

35. 1. "" (пустая строка) 2. HEXLET 3. циклы подходят для работы как с числами, так и со строками

36. 1. В теле цикла можно писать любые инструкции, в том числе и конструкции с условиями

2. можно вернуть значение по условию и остановить цикл 3. С помощью условий в цикле можно оптимизировать работу программы 4. С помощью if

37.1 x -= 1

x--

2.   
эти операции имеют побочные эффекты

в JavaScript код с этими операциями не пройдет проверку линтера

их нужно использовать с осторожностью

3. 6 **4**. 9, 8, 7, 6 **5.** Hlo

38.1. да 2. цикл for может быть бесконечным.. цикл while может быть бесконечным

3. Функция getHalfString() принимает строку и возвращает половину строки (если длина строки нечетная, то возвращается большая часть). Добавьте недостающие части кода:

Начало формы

const getHalfString = (str) => {

let result = '';

for (let i = 0; i < .length / 2; i += ) {

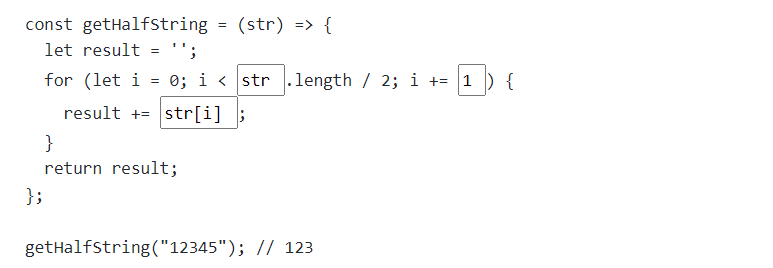
result += ;

}

return result;

};

getHalfString("12345"); // 123

Конец формы