26. Реализуйте функцию capitalize(), которая принимает непустую строку и приводит первую букву строки к верхнему регистру:

|  |  |
| --- | --- |
|  | const name = 'arya'; |
|  | console.log(capitalize(name)); *// => "Arya"* |

Чтобы получить подстроку (или символ) из строки, используйте метод [slice()](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/String/slice):

'welcome'.slice(2, 5); *// "lco"*

Для приведения строки к верхнему регистру используйте метод [toUpperCase()](https://developer.mozilla.org/ru/docs/Web/JavaScript/Reference/Global_Objects/String/toUpperCase)

'welcome'.toUpperCase(); *// "WELCOME"*

// BEGIN (write your solution here)

capitalize=()=>{

}

// END

27. Реализуйте функцию isInternationalPhone(), которая проверяет формат указанного телефона. Если телефон начинается с *+*, значит это международный формат.

|  |  |
| --- | --- |
|  | isInternationalPhone('89602223423'); *// false* |
|  | isInternationalPhone('+79602223423'); *// true* |

28. Реализуйте функцию isLeapYear(), которая определяет, является ли год високосным или нет. Год будет високосным, если он кратен (то есть делится без остатка) 400 или он одновременно кратен 4 и не кратен 100. Как видите, в определении уже заложена вся необходимая логика, осталось только переложить её на код:

|  |  |
| --- | --- |
|  | isLeapYear(2018); *// false* |
|  | isLeapYear(2017); *// false* |
|  | isLeapYear(2016); *// true* |

Кратность можно проверять так:

|  |  |
| --- | --- |
|  | *// % - возвращает остаток от деления левого операнда на правый* |
|  | *// Проверяем что number кратен 10* |
|  | number % 10 === 0 |
|  |  |
|  | *// Проверяем что number не кратен 10* |
|  | number % 10 !== 0 |

29. Реализуйте функцию getLetter(), которая извлекает из переданной строки указанный символ (по порядковому номеру, а не индексу) и возвращает его наружу. Если такого символа нет, то функция возвращает пустую строку.

Примеры вызова:

|  |  |
| --- | --- |
|  | const name = 'Hexlet'; |
|  |  |
|  | *// Обычное обращение возвращает undefined* |
|  | name[10]; *// undefined* |
|  |  |
|  | *// 11 символ соответствует 10 индексу* |
|  | getLetter(name, 11); *// ''* |
|  |  |
|  | getLetter(name, 1); *// 'H'* |
|  | getLetter(name, 0); *// ''* |
|  | getLetter(name, 6); *// 't'* |

30. normalizeUrl.js

Реализуйте функцию normalizeUrl(), которая выполняет так называемую нормализацию данных. Она принимает адрес сайта и возвращает его с *https://* в начале.

Функция принимает адреса в виде *АДРЕС* или *https://АДРЕС*, но всегда возвращает адрес в виде *https://АДРЕС*

Можно использовать метод [startsWith()](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/String/startsWith), чтобы проверить, начинается ли строка с префикса *https://*. А потом на основе этого добавлять или не добавлять *https://*.

Примеры вызова:

|  |  |
| --- | --- |
|  | normalizeUrl("google.com"); *// "https://google.com"* |
|  | normalizeUrl("https://ai.fi"); *// "https://ai.fi"* |

31. Реализуйте функцию convertText(), которая принимает на вход строку и, если первая буква не заглавная, возвращает перевернутый вариант исходной строки. Если первая буква заглавная, то строка возвращается без изменений. Если на вход передана пустая строка, функция должна вернуть пустую строку.

Примеры вызова:

|  |  |
| --- | --- |
|  | convertText('Hello'); *// 'Hello'* |
|  | convertText('hello'); *// 'olleh'* |
|  |  |
|  | *// Не забудьте учесть пустую строку* |
|  | convertText(''); *// ''* |

Перевернуть строчку можно используя функцию reverse(). В качестве аргумента в неё нужно передать строку, которую мы хотим перевернуть:

|  |  |
| --- | --- |
|  | const result = reverse('Hello!'); |
|  | console.log(result); *// => '!olleH'* |

Есть разные подходы к решению этой задачи. Возможно, вам пригодится метод [toUpperCase()](https://developer.mozilla.org/ru/docs/Web/JavaScript/Reference/Global_Objects/String/toUpperCase) и возможность получения символа из строки (например, str[0]).

Попробуйте написать два варианта функции: с обычным if-else, и с тернарным оператором.

Подсказки

* Подумайте какую проверку нужно написать первым делом: проверку заглавности буквы или является ли пустой строка. Что первичнее?

import reverse from './reverse';

// BEGIN (write your solution here)

// END

export default convertText;

32. Реализуйте функцию getNumberExplanation(), которая принимает на вход число и возвращает объяснение этого числа. Если для числа нет объяснения, то возвращается just a number. Объяснения есть только для следующих чисел:

|  |  |
| --- | --- |
|  | \* 666 - devil number |
|  | \* 42 - answer for everything |
|  | \* 7 - prime number |

Примеры вызова функции:

|  |  |
| --- | --- |
|  | getNumberExplanation(8); *// just a number* |
|  | getNumberExplanation(666); *// devil number* |
|  | getNumberExplanation(42); *// answer for everything* |
|  | getNumberExplanation(7); *// prime number* |

33. Модифицируйте функцию printNumbers() так, чтобы она выводила числа в обратном порядке. Для этого нужно идти от верхней границы к нижней. То есть счётчик должен быть инициализирован максимальным значением, а в теле цикла его нужно уменьшать до нижней границы.

printNumbers(4);

|  |  |
| --- | --- |
|  | 4 |
|  | 3 |
|  | 2 |
|  | 1 |
|  | finished! |

// @ts-check

/\* eslint-disable no-console \*/

const printNumbers = (initialNumber) => {

  // BEGIN (write your solution here)

  // END

};export default printNumbers;

26. ответ // BEGIN (write your solution here)

const capitalize = (text) => `${text[0].toUpperCase()}${text.slice(1)}`;

// END

export default capitalize;

27. // BEGIN

isInternationalPhone = (phone) => phone[0,1]==+7;

// END

export default isInternationalPhone;

28. // BEGIN (write your solution here)

const isLeapYear=(year)=>

  {  const result = year % 400 === 0 || (year % 4 === 0 && year % 100 !== 0);

  return result;

}

// END

29. // BEGIN

const getLetter = (text, position) => text[position - 1] || '';

// END

30. // BEGIN

const normalizeUrl = (site) => {

  let normalizedUrl;

  if (site.startsWith('https://')) {

    normalizedUrl = site;

  } else {

    normalizedUrl = `https://${site}`;

  }

  return normalizedUrl;

};

// END

export default normalizeUrl;

31. // BEGIN

const convertText = (text) => {

if (text === '') {

return '';

}

const reversable = text[0] !== text[0].toUpperCase();

return reversable ? reverse(text) : text;

};

// END

32. // BEGIN

const getNumberExplanation = (number) => {

  switch (number) {

    case 666:

      return 'devil number';

    case 7:

      return 'prime number';

    case 42:

      return 'answer for everything';

    default:

      return 'just a number';

  }

};

// END

export default getNumberExplanation;

33. // @ts-check

/\* eslint-disable no-console \*/

const printNumbers = (initialNumber) => {

  // BEGIN

    let i = initialNumber;

    while (i >= 1) {

      console.log(i);

      i = i - 1;

    }

    console.log('finished!');

  };

  // END

export default printNumbers;