Kristianstad University Sweden

Lecture 8 - JavaScript Data Types & Variables

Deema Aloom

Kristianstad University Sweden

Built-in functions

- `**parseInt**(string)`: Converts a string to an integer.
- `**parseFloat**(string)`: Converts a string to a floating-point number.
- `**isNaN**(value)`: Checks if a value is not a number (NaN).
- `**String**(variable)`: Converts a value to a string.
- `Number.**toString**()`: Converts a number to a string.

Quiz

```
let num = 10;
let str = "5";
let results = num + str;
return?
```

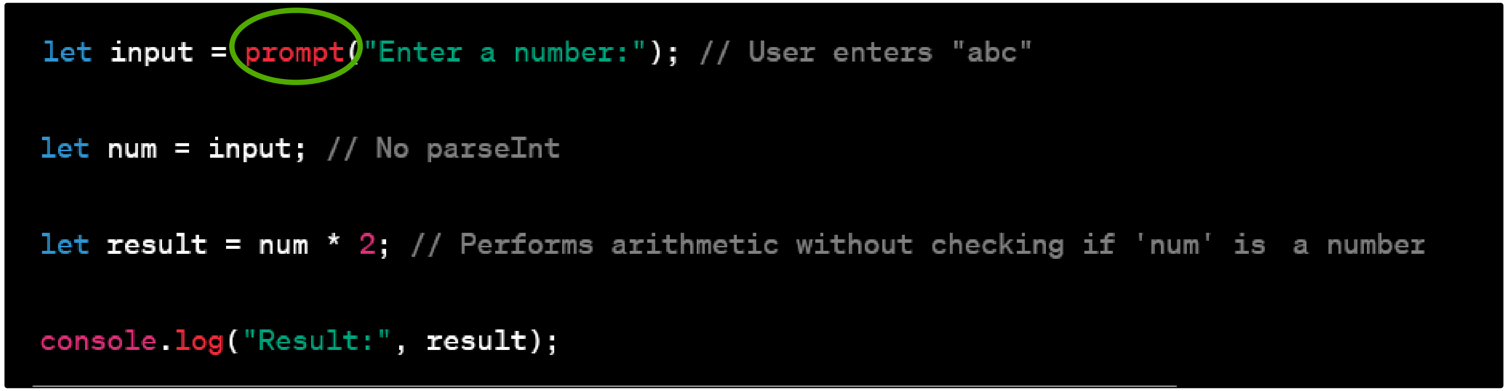
Quiz

```
let num = 10;
let str = "5";
let results = num + str;
return? "105"
```

Be cautious when mixing data types;
JavaScript may perform implicit type conversions

Common Mistakes:

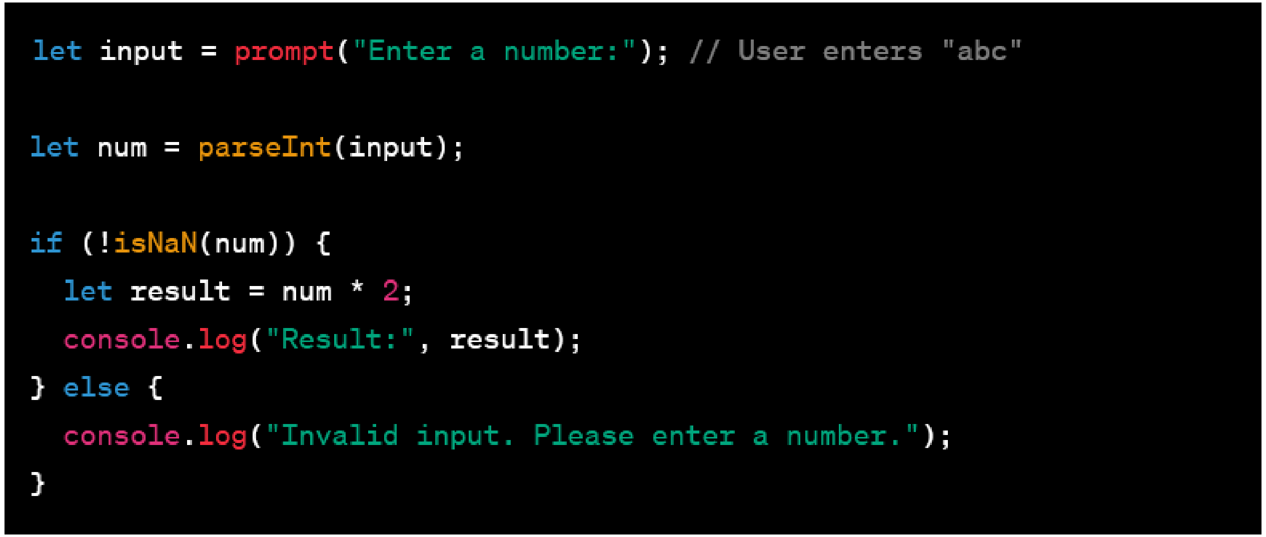
- Ensure proper handling of string inputs to avoid unexpected results in calculations.



```
let input = prompt("Enter a number:"); // User enters "abc"

let num = input; // No parseInt

let result = num * 2; // Performs arithmetic without checking if 'num' is  a number

console.log("Result:", result);
```

Common Mistakes:

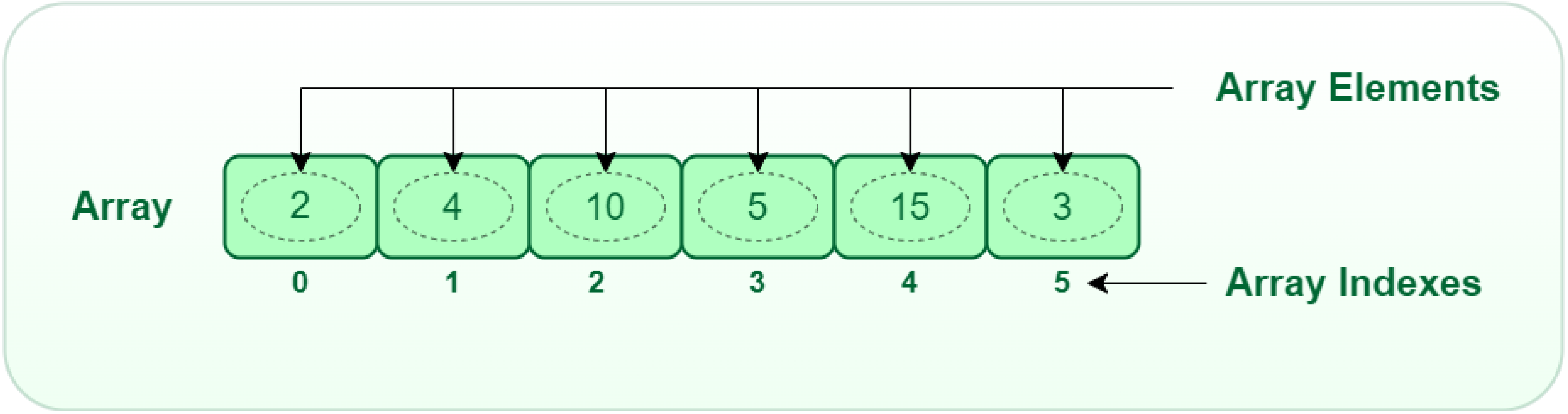
- Ensure proper handling of string inputs to avoid unexpected results in calculations.



```
let input = prompt("Enter a number:"); // User enters "abc"

let num = parseInt(input);

if (!isNaN(num)) {
  let result = num * 2;
  console.log("Result:", result);
} else {
  console.log("Invalid input. Please enter a number.");
}
```

Arrays

```
const cars = [2,4,10,5,15,3];
```



geeksforgeeks.org

Kristianstad University Sweden

Arrays

-
  ```
  const cars = ["Saab", "Volvo", "BMW"];
  ```

-
  ```
  const cars = [];
  cars[0]= "Saab";
  cars[1]= "Volvo";
  cars[2]= "BMW";
  ```

-
  ```
  const cars = new Array("Saab", "Volvo", "BMW");
  ```

Arrays

- Arrays in JavaScript can be a mixed of Data Types
- `const person = ["John", "Doe", 46];`

-
  ```
  const fruits = ["Banana", "Orange", "Apple", "Mango"];
  let fLen = fruits.length;
  ```

Kristianstad University Sweden

Arrays

- Arrays in JavaScript can be a mixed of Data Types
- `const person = ["John", "Doe", 46];`

- `const fruits = ["Banana", "Orange", "Apple", "Mango"];`

- `let fLen = fruits.length;`
- `fruits.push("Lemon");  // Adds a new element (Lemon) to fruits`

Kristianstad University Sweden

NOTE

- You should use objects when you want the element names to be strings (text).
- You should use arrays when you want the element names to be numbers.

Not a good practice!!

```
const person = [];
person["firstName"] = "John";
person["lastName"] = "Doe";
person["age"] = 46;
person.length;     // Will return 0
person[0];         // Will return undefined
```

Not a good practice!!

```
const person = [];
person["firstName"] = "John";
person["lastName"] = "Doe";
person["age"] = 46;
person.length;     // Will return 0
person[0];         // Will return undefined
```

If you use named indexes, JavaScript will redefine the array to an object. Some array methods and properties will produce **incorrect results**.

Array Methods

- **toString()** converts an array to a string of (comma separated) array values.

- The **join()** method also joins all array elements into a string.
  - It behaves just like toString(), but in addition you can specify the separator

Array Methods

- The **pop()** method removes the last element from an array.

- The **push()** method adds a new element to an array (at the end).

Array Methods

- The **shift**() method returns the value that was "shifted out":
- The **unshift**() method adds a new element to an array (at the beginning), and "unshifts" older elements:

Kristianstad University Sweden

Object

```
let person = {
    name: ['Alex', 'Bob'],
    age: 22,
    bio: function () {
        console.log(`${this.name[0]} ${this.name[1]} is 
${this.age} years old.`);
    },
    intro: function () {
        return `Hi! I’m `+ this.name[0]);
    }
};
```

Kristianstad University Sweden

Object

```
let person = {
    name: ['Alex', 'Bob'],
    age: 22,
    bio: function () {
        console.log(`${this.name[0]} ${this.name[1]} is
${this.age} years old.`);
    },
    intro: function () {
        return `Hi! I’m `+ this.name[0]);
    }
};
```

Prop Array

methods

Extend object

- `person.age`
- `person['age’]`

- `person.bio()`
- `person.intro()`

Kristianstad University Sweden

Extend object

- `person.age`
- `person['age']`

- `person.bio()`
- `person.intro()`

- `person.address ='somewhere'`
- `person.lives = function () {`
- `    return `I live in` + this.address);`

Kristianstad University Sweden

String

- A string is an object used for holding a sequence of characters
- Strings in JavaScript can be “ ” or ‘ ‘ or ` `

- https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/String

Kristianstad University Sweden

Initialize a string

- `let string1 = 'string'`
- `let string2 = new String(string1)`

- `console.log(typeof string1) // Logs "string"`
- `console.log(typeof string2)  // Logs "object"`

Kristianstad University Sweden

String Methods

- **Length is a property to find the total n. of characters**

- **concat()** and + and += string operators

- **indexOf()** //checking for the existence or location of substrings
  - It returns the first occurrence of a character or a substring in a `String`.
  - If it cannot find the character or substring, it will return -1.

- **search()** can be used either like indexof() or can use a regular expression. indexOf() is faster .

Kristianstad University Sweden

String Methods - continue

- **charAt()** //returns the character of a `String` at a specified index.
    - The index value is passed inside of the (), and should lie between 0 and length()-1.

- **toUpperCase()** //returns the string value converted to uppercase.
- **toLowerCase()** //returns the string value converted to lowercase.

Substring and substr

- The substring method returns a part of a given string.
  - **substring**`(start)`
  - **substring**`(start, end)`

- The substr method returns a part of a given string.
  - **substr** `(start)`
  - **substr** `(start, length)`

Split

- `split()` splits a string into an array of substrings, and returns the array:
  - **split**`(separator)`
  - **split**`(separator, limit)`

```
let text = “Learning Javascript string’s methods";
const myArray1 = text.split(" ");
const myArray2 = text.split(" ", 2);
```

```
Learning, Javascript, string’s, methods

Learning, Javascript
```

Kristianstad University Sweden

Trim strings

- **startsWith(substring)** //returns true or false
- **endsWith(substring)** //returns true or false
- **includes(substring)** //returns true or false

- (case-sensitive match)
- You may read further in the book …

Built-in Objects

- **Math Object:**
- The Math object provides a set of methods and properties for performing mathematical operations.
- It includes functions like Math.random() for generating random numbers, Math.round() for rounding numbers, Math.sqrt() for calculating square roots, and many more.

Built-in Objects

- **Date Object:**
- The Date object is used for working with dates and times.
- It allows you to
  - create and manipulate dates
  - perform operations like getting the current date and time, formatting dates, calculating time intervals, and more.

Kristianstad University Sweden

Questions

Kristianstad University Sweden