Text Processing

Table of Contents

Strings

What is String?

What is String?

- Strings are sequences of characters
  - Like arrays, they have **length** (access by index)
- Strings are enclosed in **three types** of quotes

```
let str = "Hello";
```

```
let str = 'Hello';
```

```
let str = `Hello`;
```

- Concatenated using the "**+**" operator

```
let s = "Hello" + " " + "JS";
```

Strings Are Immutable

- Strings are **immutable** (read-only) sequences of characters
- Accessible by **index**

```
let str = "Hello, JS";
let ch = str[2]; // Expected output: l
ch = str.charAt(2); // Expected output: l
// Both declarations are the same
```

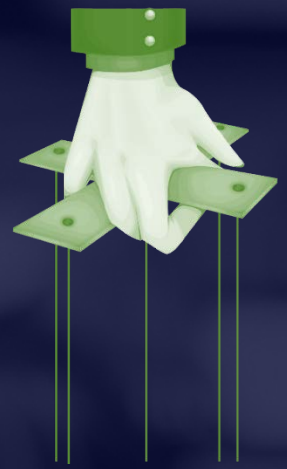
Problem: Print Characters

- Receive a **string**
- Print all the characters on separate lines

```
AWord
```

```
A
W
o
r
d
```

```
function solve(string) {
  for (let ch of string) {
    console.log(ch);
  }
}
```



Manipulating Strings

Concatenating

- Use the "+" or the "+=" operators

```
let text = "Hello" + ", ";

// Expected output: "Hello, "

text += "JS!"; // "Hello, JS!"
```

- Use the **concat()** method

```
let greet = "Hello, ";
let name = "John";
let result = greet.concat(name);
console.log(result); // Expected output: "Hello, John"
```

Searching for Substrings

- **indexOf(substr)**

```
let str = "I am JavaScript developer";
console.log(str.indexOf("Java")); // Expected output: 5
console.log(str.indexOf("java")); // Expected output: -1
```

- **lastIndexOf(substr)**

```
let str = "Intro to programming";
let last = str.lastIndexOf("o");
console.log(last); // Expected output: 11
```

Extracting Substrings

- **substr(startIndex, length)**

```
let str = "I am JavaScript developer";
let sub = str.substr(5, 10);
console.log(sub); // Expected output: JavaScript
```

- **substring(startIndex, endIndex?)**

```
let str = "I am JavaScript developer";
let sub = str.substring(5, 10);
console.log(sub); // Expected output: JavaS
```

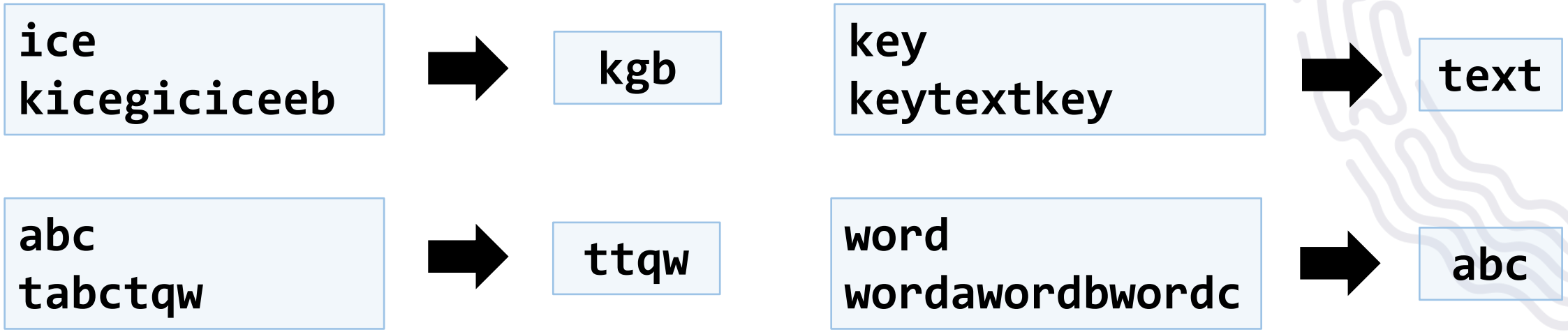
String Operations

- **replace(search, replacement)**

```
let text = "Hello, john@kingsland.us, you have been
using john@kingsland.us in your registration.";

let replacedText = text.replace(".us", ".com");

console.log(replacedText);
// Hello, john@kingsland.com, you have been using
john@kingsland.us in your registration.
```

Problem: Remove Occurrences

- Receive a **text** and a **remove word**
- Remove all substrings that are equal to the remove word



Solution: Remove Occurrences

```
function solve(word, text) {
  let old;
  while (old !== text) {
    old = text;
    text = text.replace(word, '');
  }

  console.log(text);
}
```

Problem: Substring

- Receive a **string** a **start index** and **count**
- Print the **substring** of the received string

`"ASentance", 1, 8`	➡	`Sentance`
`"JavaScript", 4, 6`	➡	`Script`

Solution: Substring

```
function solve(text, startIndex, count) {
 let substring = text
    .substr(startIndex, count);

 console.log(substring);
}
```

Splitting and Finding

split(separator)

```
let text = "I love fruits";
let words = text.split(' ');
console.log(words); // Expected output: ['I', 'love', 'fruits']
```

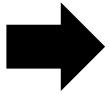
includes(substr)

```
let text = "I love fruits.";
console.log(text.includes("fruits")); // Expected output: True
console.log(text.includes("banana")); // Expected output: False
```

Repeating Strings

- **`Str.repeat(count)`** - Creates a new string repeated count times

```
let n = 3;

for(let i = 1; i <= n; i++) {

  console.log('*'.repeat(i));

}
```



```
// *

// **

// ***
```

Problem: Censored Words

- Receive a **string** and a **single word**
- Find all **occurrences** of that word in the text and **replace** them with the corresponding amount of '*'

```
A small sentence with some words,
small
```

```
A ***** sentence with some words
```

Solution: Censored Words

```
function solve(text, word) {
 while (text.includes(word)) {
   text = text.replace(word, '*'.repeat(word.length));
 }
 console.log(text);
}
```

Trimming Strings

- Use trim() method to remove whitespaces (spaces, tabs, no-break space, etc.) from both ends of a string

```
let text = "   Annoying spaces      ";

console.log(text.trim()); // Expected output: "Annoying spaces"
```

- Use trimStart() or trimEnd() to remove whitespaces only at the beginning or at the end

```
let text = "   Annoying spaces      ";

text = text.trimStart(); text = text.trimEnd();

console.log(text); // Expected output: "Annoying spaces"
```

Starts With/Ends with

- Use startsWith() to determine whether a string begins with the characters of a specified substring

```
let text = "My name is John";

console.log(text.startsWith('My')); // Expected output: true
```

- Use endsWith() to determine whether a string ends with the characters of a specified substring

```
let text = "My name is John";

console.log(text.endsWith('John')); // Expected output: true
```

Padding at the Start and End

- Use padStart() to add to the current string another substring at the start until a length is reached

Receives length and substring

```
let bitAsStr = "010";

console.log(text.padStart(8, '0')); // Expected output: 00000010
```

- Use padEnd() to add to the current string another substring at the end until a length is reached

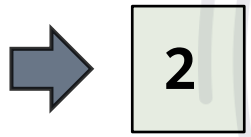
```
let sentence = "He passed away";

console.log(text.padEnd(20, '.'));

// Expected output: He passed away......
```

Problem: Count String Occurrences

- Receive a text and a word that you need to search
- Find all occurrences of that word and print them

"This is a word and it also is a sentence",

"is"



Solution: Count String Occurrences

```
function solve(text, search) {
  let words = text.split(' ');
  let counter = 0;
  for (let w of words) {
    if (w === search) {
      counter++;
    }
  }
  console.log(counter);
}
```

Live Exercises

Summary

- Strings are immutable sequences of Unicode characters
- String processing methods
  - `concat()`, `indexOf()`, `includes()`, `substring()`, `split()`, `replace()`

Questions?

License

- This course (slides, examples, demos, exercises, homework, documents, videos and other assets) is **copyrighted content**
- Unauthorized copy, reproduction or use is illegal
- © Kingsland University – https://kingslanduniversity.com


KINGSLAND
UNIVERSITY
THANK YOU