JavaScript Syntax & Use Cases Cheatsheet

Ø=ÜÌSyntax: let

Ø=ÝUse Case: Used to declare a block-scoped local variable, optionally initializing it to a value.

```
let x = 10;
```

Ø=ÜÌSyntax: const

Ø=ÝUse Case: Used to declare a block-scoped, read-only named constant.

```
const PI = 3.14;
```

Ø=ÜÌSyntax: var

Ø=ÝUse Case: Declares a variable, optionally initializing it to a value. Function-scoped or globally-scoped.

```
var name = 'John';
```

Ø=ÜÌSyntax: function

Ø=ÝUse Case: Defines a function with a specified name and parameters.

```
function greet(name) { return 'Hello ' + name; }
```

Ø=ÜÌSyntax: if...else

Ø=ÝUse Case: Executes a block of code if a specified condition is true, otherwise another block of code is executed.

```
if (x > 10) { console.log('Greater'); } else { console.log('Smaller'); }
```

Ø=ÜÌSyntax: for

Ø=ÝUse Case: Creates a loop that consists of three optional expressions, enclosed in parentheses and separated by semicolons, followed by a statement executed in the loop.

```
for (let i = 0; i < 5; i++) { console.log(i); }
```

Ø=ÜÌSyntax: while

Ø=ÝUse Case: Creates a loop that executes a specified statement as long as the test condition evaluates to true.

```
while (n < 3) { n++; }
```

Ø=ÜÌSyntax: do...while

Ø=ÝUse Case: Executes a block of code once, and then repeats the loop as long as a specified condition is true.

```
do { n++; } while (n < 3);
```

Ø=ÜÌSyntax: switch

Ø=ÝUse Case: Evaluates an expression, matching the expression's value to a case clause, and executes statements associated with that case.

```
switch (day) { case 1: console.log('Monday'); break; }
```

Ø=ÜÌSyntax: try...catch

Ø=ÝUse Case: Marks a block of statements to try, and specifies a response, should an exception be thrown.

```
try { riskyFunction(); } catch (e) { console.error(e); }
```

Ø=ÜÌSyntax: Array

Ø=ÝUse Case: Used to store multiple values in a single variable.

```
let fruits = ['Apple', 'Banana', 'Cherry'];
```

Ø=ÜÌSyntax: Object

Ø=ÝUse Case: Used to store collections of data and more complex entities.

```
let person = { firstName: 'John', lastName: 'Doe' };
```

Ø=ÜÌSyntax: Arrow Function

Ø=ÝUse Case: Provides a shorter syntax for writing function expressions

```
const add = (a, b) => a + b;
```