Alexander Schalk / Victoria Chmielewski 7/15/2019

The arrow operator introduces a way of writing a concise functions.

Test Sample(s):

<!doctype html>

<html>

<head>

<title>Example of Arrow Operator</title>

</head>

<body>

<h1>Example of Arrow Operator</h1>

<p>Use the browser’s debugger to view the js executing</p>

</body>

<p id="demo"></p>

<script>

// Create objects:

var isHot = (item) => "Bring a " + item;

//var isCold = (item) => "Bring a " + item;

//var isWet = (item) => "Bring a " + item;

// Display some data from the object:

document.getElementById("demo").innerHTML = isHot("fan");

//document.getElementById("demo").innerHTML = isCold("coat");

//document.getElementById("demo").innerHTML = isWet("rain jacket");

debugger;

</script>

</html>

Features of the arrow operator:

* No Parameters: Parenthesis are when there are no parameters
* You can return object literal expression
* You can use the arrow function for array manipulation
* Arrow functions reduce the confusion of surrounding “this” keyword
* {} are not required if there is only one expression

Pro and Cons:

Cons

* Harder to debug: Will not be able to trace any kind of error
* No self-referencing: Will not have any kind of recursion, event handler that needs to unbind
* Object Methods
* Cannot be used as a constructor
* Cannot be used as Generator

Pros

* Shorter Syntax
* No Binding of “this”
* Less verbose to declare then a traditional function
* Eliminates some lines of code
* Implicitly returning value