

FIT2102

Programming Paradigms

Tutorial 1

Introduction to Javascript and Higher Order Functions

Faculty of Information Technology



MONASH
University

Introduction To Javascript

- Javascript is one of the top three most popular languages currently.
- Currently used widely in web development.
- Supports a range of different paradigms

The logo consists of a solid yellow square. In the bottom right corner of this square, the letters 'JS' are written in a bold, dark grey, sans-serif font.

Mutable and Immutable Variables

Mutable Variables: Defined with the *let* keyword.

```
let t = 2; // t is mutable
```

Immutable Variables: Defined with the *const* keyword

```
const z = 1; // constant (immutable variable)
```

Basic Operators

Binary Operators:

```
x % y    // modulo
x == y   // loose* equality
x != y   // loose* inequality
x === y  // strict* equality
x !== y  // strict* inequality
```

```
a && b    // logical and
a || b    // logical or
```

```
a & b     // bitwise and
a | b     // bitwise or
```

Unary Operators:

```
i++      // post-increment
++i      // pre-increment
i--      // post-increment
--i      // pre-increment
!x       // not x
```

Ternary Conditional Operator:

```
<condition> ? <true result> : <false result>
```

* Loose equality means type conversion may occur

In-place math operators:

```
x += <expr>
// add result of expr to x
// also -=, *=, /=, |=, &=.
```

If Statements

```
function maxVal(x, y) {  
  if (x >= y) {  
    return x;  
  } else {  
    return y;  
  }  
}
```

```
function maxVal(x, y) {  
  return x >= y ? x : y;  
}
```

Loops: Imperative and Recursive

```
function sumTo(n) {  
  let sum = 0;  
  for (let i = 1; i <= n; i++) {  
    sum += i;  
  }  
  return sum;  
}
```

```
function sumTo(n) {  
  return n ? n + sumTo(n-1) : 0;  
}
```

Higher Order Functions

- A higher order function is a function that takes a function as an argument, or returns a function

```
function square(x) {  
  return x * x;  
}
```

```
function sumTo(n, f) {  
  return n ? f(n) + sumTo(n-1, f) : 0;  
}
```

```
sumTo(10, square)
```

```
> 385
```