

Control Flow

Liam McLennan
www.pluralsight.com



Outline

- **Conditional control flow**
- **Iteration**
- **Error handling**

Blocks

- Group a set of statements between curly braces
- Do not provide variable scope

```
{  
  
}
```

if

- **Execute a block if a logical condition is true**

```
if (2 === (1 + 1)) {  
    // execute this block  
}
```

if ... else

- Alternative block to execute if the condition is false

```
if (false) {  
  
} else {  
    // execute this block  
}
```

switch

- Choose from a set of possibilities
- Same as C#



**EXAMP
LE**

for

```
for (var i = 0; i < 10; i++) {  
}
```

Loop initializer

Condition

Incrementer



**EXAMP
LE**

for ... in

- Iterates over the property keys of an object
- Similar to C#'s foreach



**EXAMP
LE**

while

- **Pre-tested loop**
- **Ensure that the loop condition will eventually become false**

```
while (keepGoing) {  
    keepGoing = false;  
}
```

do ... while

- Post-tested loop
- Ensure that the loop condition will eventually become false

```
do {  
    keepGoing = false;  
} while (keepGoing);
```



**EXAMP
LE**

Error Handling - throw

- Throw an exception when an unusual error condition occurs
- Exceptions are thrown using the 'throw' statement
- The exception object can be accessed when the exception is caught

```
if (somethingGoesWrong) {  
    throw {  
        name: "SomethingWentWrongError",  
        message: "Something went wrong. You should fix it."  
    };  
}
```

Error Handling - catch

- When an exception is thrown within a 'try' block it can be caught and handled within a 'catch' block.
- A 'finally' block can be used to guarantee execution of some statements, even in the event of an exception

```
try {  
    throw {  
        name: "SomethingWentWrongError",  
        message: "Something went wrong. You should fix it."  
    };  
} catch (e) {  
    // handle the exception here  
}  
finally {  
    // this is executed even if an exception occurs  
}
```



**EXAMP
LE**

Summary

- **If, else and switch**
- **For, for ... in, while and do ... while**
- **Exception handling with try, catch & finally**

For more in-depth **online** developer **training** visit



on-demand content from authors you **trust**

