

Chapter 8

How to code control statements

The equality operators

Operator	Description
==	Equal
!=	Not equal

The identity operators

Operator	Description
===	Equal
!==	Not equal

Description

- The *equality operators* perform *type coercion*.
- The *identity operators* do not perform type coercion.

The relational operators

Operator	Description
<	Less than
<=	Less than or equal
>	Greater than
>=	Greater than or equal

Comparing strings to numbers

Expression	Result
1 < "3"	true
"10" < 3	false

Comparing strings to strings

Expression	Result
"apple" < "orange"	true
"apple" < "appletree"	true
"Orange" < "apple"	true
"@" < "\$"	false

The logical operators

Operator	Name
!	NOT
&&	AND
	OR

The NOT operator

```
!isNaN(number)
```

The AND operator

```
age >= 18 && score >= 680
```

The OR operator

```
state == "CA" || state == "NC"
```

The order of precedence

Order	Operators
1	!
2	<, <=, >, >=
3	==, !=, ===, !==
4	&&
5	

AND, OR, and NOT operators

```
!oldCustomer || loanAmount >= 10000 &&  
    score < minScore + 200
```

How parentheses can change the evaluation

```
(!oldCustomer || loanAmount >= 10000) &&  
    score < minScore + 200
```

An if clause with one statement and no braces

```
if ( rate === undefined ) rate = 0.075;
```

An if clause with one statement and braces

```
if ( qualified ) {  
    alert("You qualify for enrollment.");  
}
```

If and else clauses with no braces

```
if ( age >= 18 )  
    alert("You may vote.");  
else  
    alert("You may not vote.");
```

Why you should use braces

```
if ( age >= 18 )  
    alert("You may vote.");  
else  
    alert("You may not vote.");  
    may_vote = false;    // Not a part of the else clause.
```

Braces make your code easier to modify

```
if ( score >= 680 ) {  
    alert("Your loan is approved.");  
}  
else {  
    alert("Your loan is not approved.");  
}
```

An if statement with one else if clause

```
if ( age < 18 ) {  
    alert("You're too young for a loan.");  
}  
else if ( score < 680 ) {  
    alert("Your credit score is too low for a loan.");  
}
```

An if statement with an else if and an else clause

```
if ( age < 18 ) {  
    alert("You're too young for a loan.");  
}  
else if ( score < 680 ) {  
    alert("Your credit score is too low for a loan.");  
}  
else {  
    alert("You're approved for your loan.");  
}
```

The if statement for a Future Value application

```
// Test if input is valid
if (isNaN(investment) || investment <= 0) {
    alert("Investment is not a valid number.");
} else if (isNaN(annualRate) || annualRate <= 0) {
    alert("Annual rate is not a valid number.");
} else if (isNaN(years) || years <= 0) {
    alert("Years is not a valid number.");

// If input is valid, calculate future value
} else {
    // code that calculates the future value goes here
}
```

How to use a flag to get the same results

```
// Test if input is valid
var valid = true;
if (isNaN(investment) || investment <= 0) {
    alert("Investment is not a valid number.");
    valid = false;
} else if (isNaN(annualRate) || annualRate <= 0) {
    alert("Annual rate is not a valid number.");
    valid = false;
} else if (isNaN(years) || years <= 0) {
    alert("Years is not a valid number.");
    valid = false;
}

// If input is valid, calculate the future value
if ( valid ){
    // code that calculates the future value goes here
}
```

A while loop to validate user input

```
var value = parseInt(
    prompt("Please enter a number from 1 to 10") );

while ( isNaN(value) || value < 1 || value > 10 ) {
    alert("You did not enter a number between 1 and 10.");
    value = parseInt(
        prompt("Please enter a number from 1 to 10") );
}
```

A while loop that counts dice rolls until a six is rolled

```
var rolls = 1;
while ( random_number(1,6) != 6 ) {
    rolls++;
}

alert("Number of times to roll a six: " + rolls);

// NOTE: See figure 7-5 for the random_number function
```

A while loop that finds the average of a series of numbers

```
alert("Enter a non-number to stop.");

var total = 0, count = 0, number;
number = parseFloat( prompt("Enter a number") );
while ( !isNaN(number) ) {
    total += number;
    count++;
    number = parseFloat( prompt("Enter another number") );
}

var average = total / count;

if ( isNaN(average) ) {
    alert("You didn't enter any numbers.");
} else {
    alert("The average is: " + average);
}
```

Nested while loops that find the average and max to roll a six

```
var total = 0, count = 0, max = -Infinity;
var rolls;

while ( count < 10000 ) {
    rolls = 1;
    while ( random_number(1, 6) != 6 ) {
        rolls++;
    }
    total += rolls;
    count++;
    if ( rolls > max ) max = rolls;
}

var average = total / count;

alert ("Average rolls: " + average);
alert ("Max rolls: " + max);
```

A do-while loop to validate user input

```
var value, valid;
do {
    value = parseInt(
        prompt("Enter a number between 1 and 10") );

    if (isNaN(value) || value < 1 || value > 10) {
        alert("You did not enter a valid number.");
        valid = false;
    } else {
        valid = true;
    }
} while ( !valid );
```

A do-while loop that counts dice rolls until a six is rolled

```
var rolls = 0;
do {
    rolls ++;
} while ( random_number(1,6) != 6 );

alert("Number of times to roll a six: " + rolls);

// NOTE: See figure 7-5 for the random_number function
```

A do-while loop that finds max and min values

```
alert("Enter a non-number to stop.");

var max = -Infinity, min = Infinity, number;
var value_entered = false, stop = false;

do {
    number = parseFloat( prompt("Enter a number") );
    if ( isNaN(number) ) {
        stop = true;
    } else {
        value_entered = true;
        if ( number > max ) max = number;
        if ( number < min ) min = number;
    }
} while ( !stop );

if (value_entered) {
    alert("Max: " + max + ", Min: " + min);
} else {
    alert("No numbers entered.");
}
```

A for statement

```
for ( var count = 1; count <= 10; count++ ) {  
    alert ( count );  
}
```

A while statement that does the same thing

```
var count = 1;  
while ( count <= 10 ) {  
    alert ( count );  
    count++;  
}
```

A for loop to display even numbers from 2 to 10

```
for ( var number = 2; number <= 10; number += 2 ) {  
    alert( number );  
}
```

A for loop to reverse a string

```
var message = "JavaScript", reverse = "";  
for (var i = message.length - 1; i >= 0; i-- ) {  
    reverse += message.charAt(i);  
}  
alert(reverse); // Displays "tpircSavaJ"
```

A for loop to display all the factors of a number

```
var number = 18;  
for ( var i = 1; i < number; i++ ) {  
    if ( number % i == 0 ) {  
        alert( i + " is a factor of " + number );  
    }  
}
```

A for loop to determine if a number is prime

```
var number = 31, prime = true;

for ( var i = 2; i < number; i++ ) {
    if ( number % i == 0 ) prime = false;
}

if (prime) {
    alert( number + " is prime.");
} else {
    alert( number + " is not prime.");
}
```

The break statement in a while loop

```
var number;
while (true) {
    number = parseInt(
        prompt("Enter a number from 1 to 10.") );

    if ( isNaN(number) || number < 1 || number > 10 ) {
        alert("Invalid entry. Try again.");
    } else {
        break;
    }
}
```

The break statement in a for loop

```
var number = 31, prime = true;
for ( var i = 2; i < number; i++ ) {
    if ( number % i == 0 ) {
        prime = false;
        break;
    }
}
```

The continue statement in a for loop

```
for ( var number = 1; number <= 10; number++ ) {  
    if ( number % 3 == 0 ) continue;  
    alert(number);  
}  
// Only displays 1, 2, 4, 5, 7, 8, and 10
```

The continue statement in a while loop

```
var number = 1;  
while ( number <= 10 ) {  
    if ( number % 3 == 0 ) {  
        number++;  
        continue;  
    }  
    alert(number);  
    number++;  
}  
// Only displays 1, 2, 4, 5, 7, 8, and 10
```