Binary

There are **10** types of people in the world:
Those who understand binary, and those who don't.

Decimal Numbers

In our daily lives, we use decimal numbers.

Decimal numbers are made from 10 digits: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9

A 1 digit decimal number: 5   = 5

A 2 digit decimal number: 23   = 20 + 3

A 3 digit decimal number: 163 = 100 + 60 + 3

Decimal Numbers

163

hundreds   tens   ones

Decimal Numbers

10^2	10^1	10^0
1	6	3

$$(1 * 10^2) + (6 * 10^1) + (3 * 10^0)$$

$$100 \quad + \quad 60 \quad + \quad 3 \quad = 163$$

Why Binary?

Computers are made from electrical components. Having only 2 distinct states provides reliability and simpler circuitry.

A **1 or 0** is used to represent the presence of voltage (or not) or a switch being **on or off**. Using multiple switches we can store larger numbers, music, photos, etc.
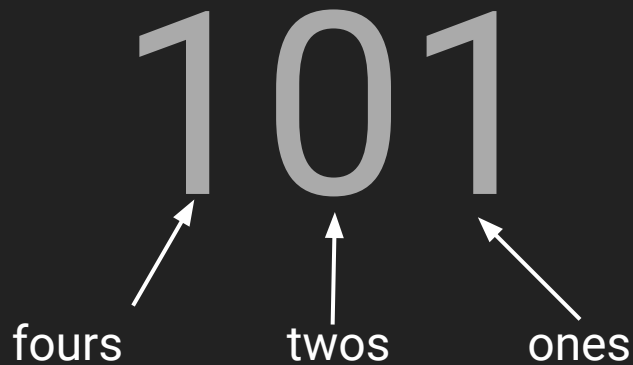
What is a Binary Number?

Binary numbers are made from 2 digits: 0, 1

Bit is short for binary digit.

Multiple bits are used to form numbers.

Decimal	Binary
0	0
1	1
2	10
3	11
4	100
5	101
6	110
7	111
8	1000
9	1001
10	1010

Binary Numbers

101

fours   twos   ones

The number 5 in binary.

Binary Numbers

2^2	2^1	2^0
1	0	1

$$(1 * 2^2) \; + \; (0 * 2^1) \; + \; (1 * 2^0)$$

$$4 \quad + \quad 0 \quad + \quad 1 \quad = 5$$

Decimal vs. Binary

Decimal is based on powers of 10

10^2	10^1	10^0
1	1	1

Decimal: (100 + 10 + 1) = 111

Binary is based on powers of 2

2^2	2^1	2^0
1	1	1

Binary: (4 + 2 + 1) = 7

Convert Binary to Decimal

Binary	Decimal
0010	?
1010	?
1100	?
1001	?

Convert Binary to Decimal

Binary	Decimal
0010	2
1010	10
1100	12
1001	9

Convert Decimal to Binary

Decimal	Binary
1	?
5	?
10	?
13	?

Examples

Decimal	Binary
1	1
5	101
10	1010
13	1101

Bits and Bytes

1 byte = 8 bits

1 kilobyte = 1024 bytes

1 megabyte = 1024 kilobytes

1 gigabytes = 1024 megabytes
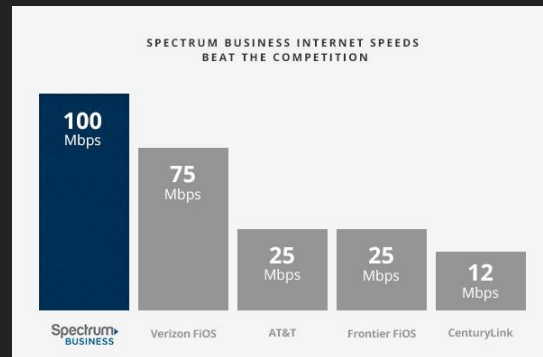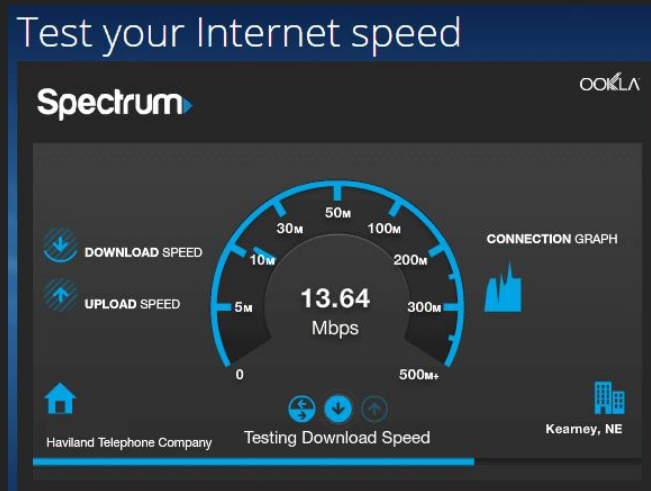
1 terabyte = 1024 gigabytes

Bits in the Real World

Game Consoles: 8 bit, 16 bit, 64 bit

Colors on Computers 8 bit, 16 bit, 24 bit

Music/Sound: 8 bit, 16 bit, 24 bit

Mbps vs MBps - Lookout for this!
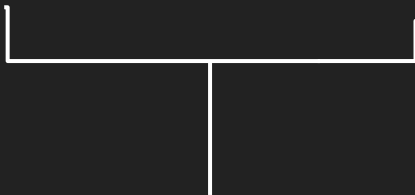
Strings

`"Hello, Again!"`

Strings

A **string literal** is a sequence of characters surrounded by quotes.

```
"Carmine"
"This class is awesome."
"42"


System.out.println("Hello, World!");
```

string literal

Data Types: int and String

int is a **primitive** type which stores negative and positive whole numbers.
String is a **class** used to store a **sequence of characters**.

```
int count = 64;

String name = "Carmine";
```

Notice the capital letter S

64 and "64" behave differently when added, printed, etc.

Assigning Values to Strings

Assigning values is similar to int. Our variable is on the left with something to evaluate on the right.

```
String name = "Carmine";

String response = "His name is Professor " + name;

name = "Professor " + name;
```

String Concatenation

Strings can be added (concatenated) together.
Adding an int to a string will (temporarily) convert the int to a string and then add it to the end.

```
String name = "Carmine";

String response = "His name is Professor " + name;

int year = 2019;

System.out.println("The year is " + year);
```

Using Scanner to get String values

Using the Scanner class, we can get a string from input.

```java
String firstName;

System.out.print("What is your first name: ");

firstName = scan.next();
```

Using Scanner to get String values

Be careful! next() looks for something separated by white-space (such as a space, tab, return, etc.

```
String fullName;

System.out.print("Enter your full name: ");

fullName = scan.next();
```

This won't behave as intended!
If you enter "Carmine Guida" ... fullName will just be "Carmine"

Using Scanner to get String values

nextLine() will look for the end of a line

```
String fullName;

System.out.print("Enter your full name: ");

fullName = scan.nextLine();
```

Using Scanner to get String values

```
firstName = scan.next();
lastName = scan.next();
System.out.print(lastName);
```

What will be the output if we **input** the following?

1. Hello World

2. Carmine Guida

3. Pace
   University

Using Scanner to get String values

```
fullName = scan.nextLine();
System.out.print(fullName);
```

What will be the output if we **input** the following?

1. Hello World

2. Carmine Guida

3. Pace
   University

Characters

'A'

Characters

A character (char) is surrounded by single quotes.

```
'A'           - Capital letter A
'!'           - Exclamation point!
'\t'          - The tab character.
```

Assigning Values to char

Assigning a value to a char is similar to int and String.

```
char initial = 'C';

char dayOfWeek = 'M';

char separator = ',';
```

Concatenating

You can add chars to the end of strings.

```
char first = 'C';
char middle = 'T';
char last = 'G';

System.out.println("Initials: " + first + middle + last);
```

charAt

You can get the character at a position in a string. charAt() is a function that strings have. Notice the index into the string starts with 0!

```
String name = "Carmine";
char initial = name.charAt(0);

System.out.println("First Initial = " + initial);
```

0	1	2	3	4	5	6
C	a	r	m	i	n	e

Using Scanner to get char values

You can use **next**() to get a String from Scanner and then use **charAt**() to get the first character.

```
char initial;

System.out.print("Enter your first initial: ");

initial = scan.next().charAt(0);
```

Escape Sequence

A character starting with a backslash \ is an escape sequence. Here are some you may need in your programs.

```
\t   tab
\n   newline
\r   carriage return
\'   single quote character
\"   double quote character
\\   backslash character
```

Data Types: int and String and char

int and char are both primitive types.
String is a class.

```
int count;

String name;

char grade;
```

↑

Notice lower case (like int)

Let's Code

Don't Forget!
Check the syllabus / schedule for reading assignments and due dates!

Floating-point Numbers

double gpa = 4.0;

Floating-point Numbers

A floating point number is a number with a decimal point. "float" refers to the fact the decimal point can be anywhere (float) in the number.

```
2.5
4.0
0.23
3.14159
98.6
12345.6789
```

Data Types: double

double is a **primitive** type you can use to store floating point numbers.

```
double gpa = 4.0;
double temperature = 98.6;
```

Assigning Values to Doubles

Assigning values is similar to int. Our variable is on the left with something to evaluate on the right.

```
double score = 90.0;


double average = 125.23 / 5.0;
```

Scientific notation: 3.0×10^8
```
double SPEED_OF_LIGHT = 3.0e8;
```

Using Scanner to get double values

Using the Scanner class, we can get a double from input.

```java
double gpa;

System.out.print("What is your GPA: ");

gpa = scan.nextDouble();
```

Our Data Type Family (So far)

```
int gpa = 4;

double gpa = 4.0;

char gpa = 'A';

String gpa = "4.0 (A)";
```

New! ⟶

Our Data Type Family (So far)

Data type types and their usage.

```
Primitive
int       Integers (negative or positive whole numbers)
double    Floating point numbers (very precise!)
char      Single character

Class
String    Sequence of characters
```

What's going to happen?

```
5 / 2                          ?

5 % 2                          ?

5.0 / 2.0                      ?

5.0 / 2                        ?

5 / 2.0                        ?
```

What's going to happen?

5 / 2	2
5 % 2	1
5.0 / 2.0	2.5
5.0 / 2	2.5
5 / 2.0	2.5

What's going to happen?

5 + 2   ?

5.0 + 2.0   ?

5.0 + 2   ?

5 + 2.0   ?

What's going to happen?

```
5 + 2                              7

5.0 + 2.0                          7.0

5.0 + 2                            7.0

5 + 2.0                            7.0
```

What's going to happen?

```
5 / 2 + 2                    ?

5 / 2 + 2.0                  ?

5 / 2.0 + 2.0                ?

2 + 5 / 2.0                  ?
```

What's going to happen?

```
5 / 2 + 2                          4

5 / 2 + 2.0                        4.0

5 / 2.0 + 2.0                      4.5

2 + 5 / 2.0                        4.5
```

That looks like some **good to know** tricky stuff!

Let's Code

Don't Forget!
Check the syllabus / schedule for reading assignments and due dates!