Notes on Binary

DAT-1.A.1
Data values can be stored in variables, lists of items, or standalone constants and can be passed as input to (or output from) procedures.

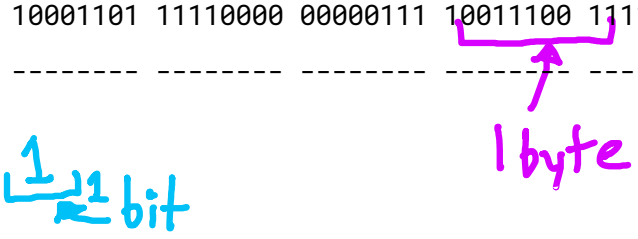
DAT-1.A.2
Computing devices represent data digitally, meaning that the lowest-level components of any value are bits.

DAT-1.A.3
Bit is shorthand for *binary digit* and is either 0 or 1.

DAT-1.A.4
A *byte* is 8 bits.

Example of a binary number:

```
10001101 11110000 00000111 10011100 11111111
-------- -------- -------- -------- --------
```



DAT-1.C.1
Number bases, including binary and decimal, are used to represent data.

DAT-1.C.2
Binary (base 2) uses only combinations of the digits zero and one.

DAT-1.C.3
Decimal (base 10) uses only combinations of the digits 0 – 9.

So binary is base 2 because there are 2 possible symbols.

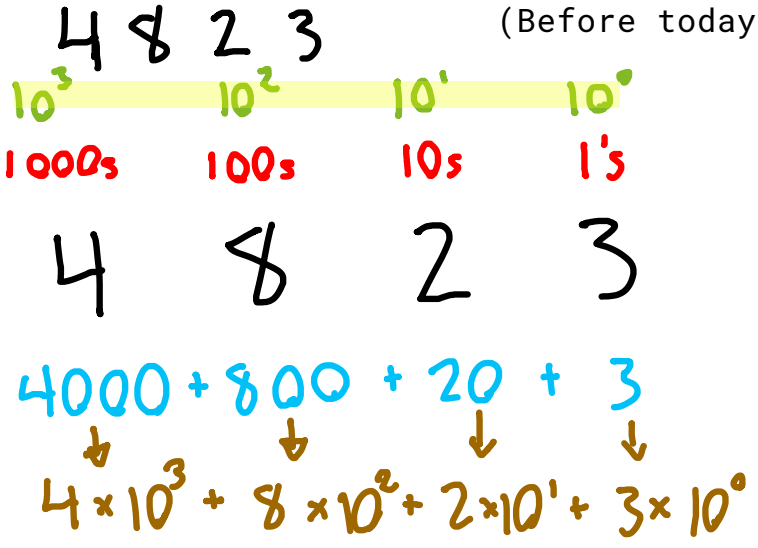
0 1

So binary is base 10 because there are 10 possible symbols.

0 1 2 3 4 5 6 7 8 9

Thinking about how base 10 numbers work...

Base 10 numbers are likely the only numbers you have ever known... :o (Before today of course!)



Converting binary numbers to base 10.

DAT-1.C

For binary numbers:

a. Calculate the binary (base 2) equivalent of a positive integer (base 10) and vice versa. **2.B**

1011

2^3 2^2 2^1 2^0

8 4 2 1

1 0 1 1

8 + 0 + 2 + 1 = 11

1×2^3 0×2^2 1×2^1 1×2^0

1011 (binary, base 2) = 11 (base 10)

shorter way of saying the same thing $1011_2 = 11_{10}$

110011

2^5 2^4 2^3 2^2 2^1 2^0

32 16 8 4 2 1

1 1 0 0 1 1 = 51 (base 10)

32 + 16 + 0 + 0 + 2 + 1

1×2^5 1×2^4 0×2^3 0×2^2 1×2^1 1×2^0

$110011_2 = 51_{10}$

---

128 64 32 16 8 4 2 1

$11001010_2 = 202_{10}$

128 + 64 + 0 + 0 + 8 + 0 + 2 + 0

256 128 64 32 16 8 4 2 1

$100011001_2 = 281_{10}$

256 + 0 + 0 + 0 + 16 + 8 + 0 + 0 + 1

128 64 32 16 8 4 2 1

$11111111_2 = 255_{10}$

128 + 64 + 32 + 16 + 8 + 4 + 2 + 1

256

$100000000_2 = 256_{10}$

Converting base 10 numbers to binary.

54_{10}

DAT-1.C

For binary numbers:

a. Calculate the binary (base 2) equivalent of a positive integer (base 10) and vice versa. **2.B**

64 32 16 8 4 2 1

0 1 1 0 1 1 0

32+16 +0 +4 +2 +0

54-32

22-16

6-0

6-4

2-2

0

$54_{10} = 110110_2$

$89_{10} = 1011001_2$

0 1 0 1 1 0 0 1

64 +0 +16+8 +0+0+1 ✓

89-64

25-0

25-16

9-8

1

$297_{10} = 100101001_2$

512 256 128 64 32 16 8 4 2 1

0 1 0 0 1 0 1 0 0 1

256 +32 +8 +1

41 9 1

$136_{10} = 10001000_2$

256 128 64 32 16 8 4 2 1

0 1 0 0 0 1 0 0 0

128 +8

8

Let's list out several binary numbers starting with 0.

b. Compare and order binary numbers. **2.B**

Patterns Noticed

Abstraction

DAT-1.A.5

Abstraction is the process of reducing complexity by focusing on the main idea. By hiding details irrelevant to the question at hand and bringing together related and useful details, abstraction reduces complexity and allows one to focus on the idea.

DAT-1.A.6

Bits are grouped to represent abstractions. These abstractions include, but are not limited to, numbers, characters, and color.

DAT-1.A.7

The same sequence of bits may represent different types of data in different contexts.

DAT-1.A.8

Analog data have values that change smoothly, rather than in discrete intervals, over time. Some examples of analog data include pitch and volume of music, colors of a painting, or position of a sprinter during a race.

DAT-1.A.9

The use of digital data to approximate real-world analog data is an example of abstraction.

DAT-1.A.10

Analog data can be closely approximated digitally using a *sampling technique*, which means measuring values of the analog signal at regular intervals called *samples*. The samples are measured to figure out the exact bits required to store each sample.

DAT-1.B.1

In many programming languages, integers are represented by a fixed number of bits, which limits the range of integer values and mathematical operations on those values. This limitation can result in overflow or other errors.

DAT-1.B.2

Other programming languages provide an abstraction through which the size of representable integers is limited only by the size of the computer's memory; this is the case for the language defined in the exam reference sheet.

DAT-1.B.3

In programming languages, the fixed number of bits used to represent real numbers limits the range and mathematical operations on these values; this limitation can result in round-off and other errors. Some real numbers are represented as approximations in computer storage.

EXCLUSION STATEMENT (EK DAT-1.B.3):

Specific range limitations for real numbers are outside the scope of this course and the AP Exam.

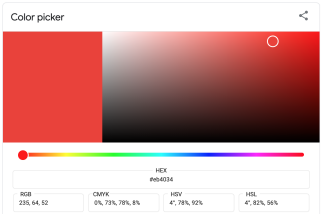
Examples of Abstractions

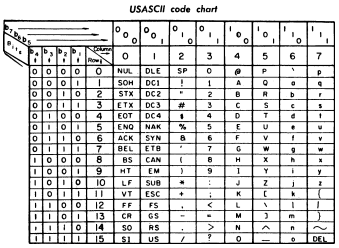
Coffee Machine

Video Game Controller/Video Games

Javascript!!!

Bottom Line: There are intuitive ways of interacting with these things that don't require you to know the full details behind how things work behind the scenes.





#'s

(You can convert binary bits to base 10.)

Analog vs Digital Clock

Analog is exact. Digital can only ever provide an approximation of the actual time.

Digital clocks have to have some exact analog source that the data is being generated from.

Matching a color in real life (analog) to an RGB value (digital) is always going to be approximate. There are more colors than possible RGB combinations.

You have to have some method of converting the real life thing (analog) to a digital approximation. The method used requires you to take samples of the real life thing and convert those samples into digital approximations. The number of bits required to store the digital info will also be highly dependent on the situation. An RGB value requires fewer bits than the dashboard for a spaceship.

In javascript, at most an integer can be represented behind the scenes by 32 bits. So this means that the maximum possible integer you can store in a javascript variable is whatever number is represented by 32 binary 1's in a row.
11111111111111111111111111111111 (equivalent to 2^(31)-1)

Unlike in Javascript, there are programming languages in which you can have integers that require more than 32 bits to represent. In fact, there are languages where there is no limit at all. (Technically, there is only no limit in the case that you had a computer with an infinite amount of memory.)