# Information Technology for Digital Humanities

Lecture 4

Mario Verdicchio

Università degli Studi di Bergamo

Academic Year 2023-2024

# Lecture 4 (October 4 2023)

- Fundamental concepts: technology
  - Previous lecture: hardware
  - Now: how the electronic signals the hardware works with are turned into something meaningful for the human users

# Information Technology for **Digital** Humanities

## Lecture 4

Mario Verdicchio

Università degli Studi di Bergamo

Academic Year 2023-2024

# Tools for **data** processing

This lesson focuses on what data/symbols/signs can be processed using computers.

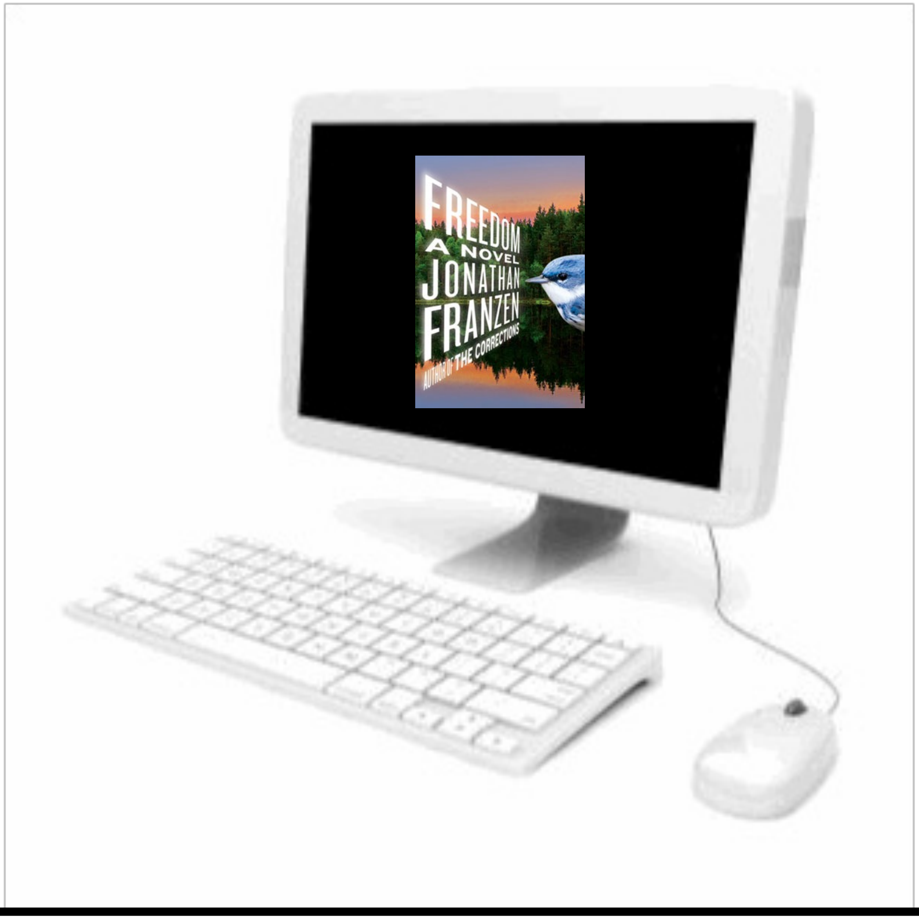
**symbol**

**8**

**symbol**

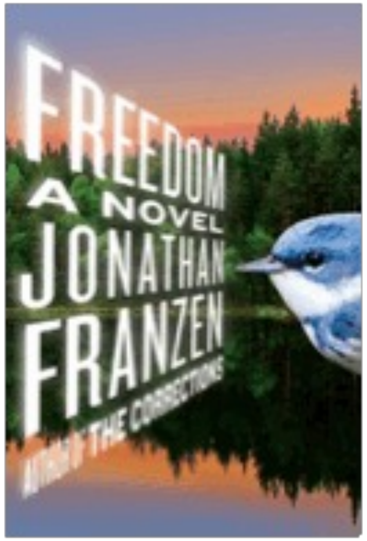
In computer science, the term “symbol” has a much weaker meaning than in the humanities: by “symbol” we mean any sign, such as a figure, without any associated meaning, as happens instead for the lion, to which we associate by symbolism the concept of “courage”.

Having simple numbers on a screen seems a bit limiting compared to what we usually see happening on the computers around us.



Authors, for example, write entire novels on their computer.

And, certainly, at least these three activities take place on the computers of the students of the course: social networks, listening to music, and watching films.

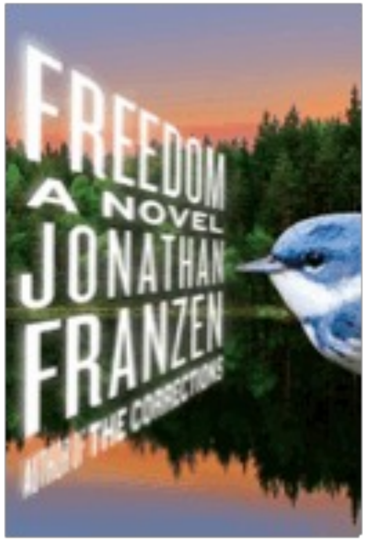


Let's think.
Premise 1: computers are nothing more than automatic executors of operations on symbols.
Premise 2: with computers we write novels, surf social networks, listen to music, watch films.

8

Conclusion:
writing novels, surfing social networks, listening to music, watching films are operations on symbols.

The reasoning is flawless, but the conclusion may leave you perplexed.



To better understand the conclusion of the previous reasoning, it is necessary to clarify the link between all these activities and the symbols processed by a computer.

8

**encoding**

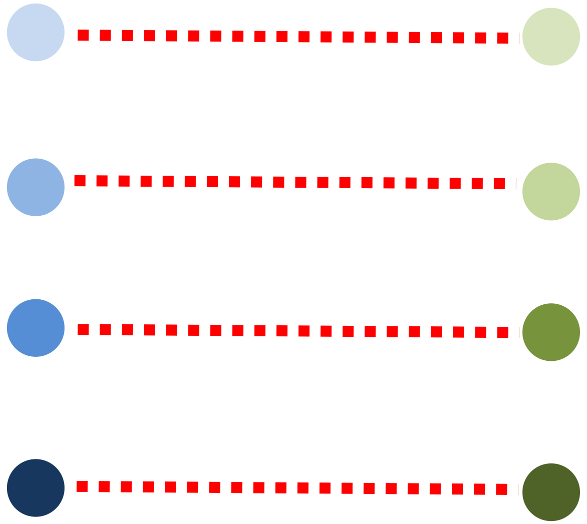
This link is given by an encoding.

# ENCODING is EVERYTHING*

*in digital technology

**ENCODING [ɪnˈkəʊdɪŋ]: biunivocal correspondence between a set of entities of any kind and a set of natural numbers.**

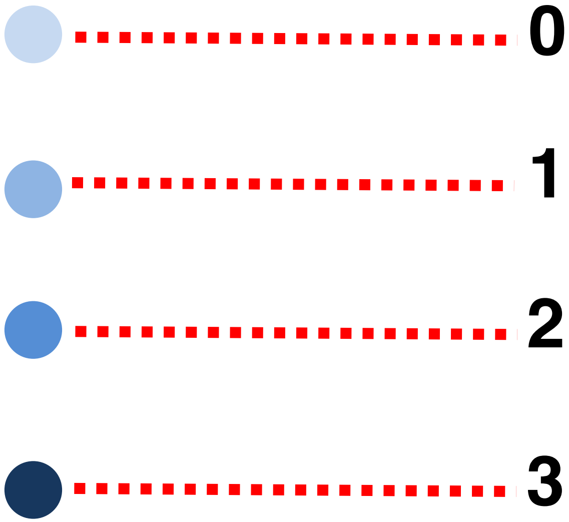
# biunivocal correspondence



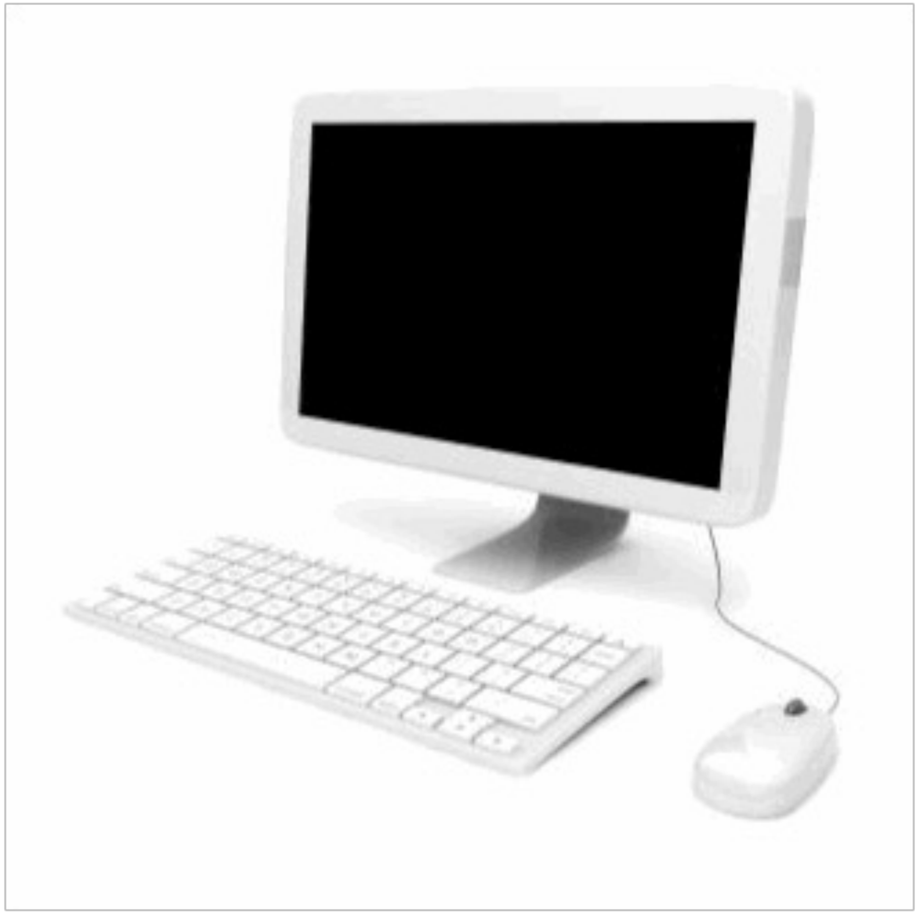
**between a set of entities of any kind**

and a set of natural numbers.

# ENCODING



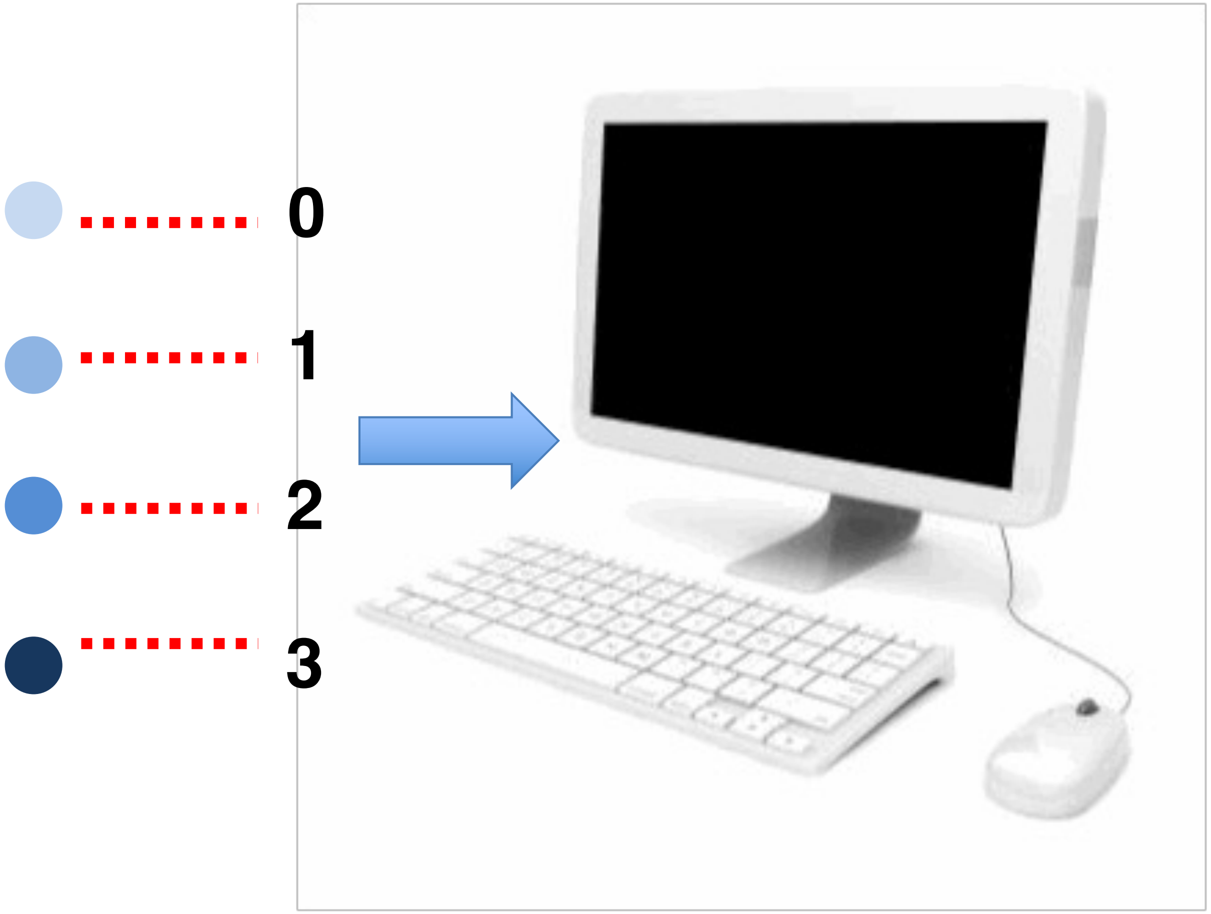
*from a conceptual perspective



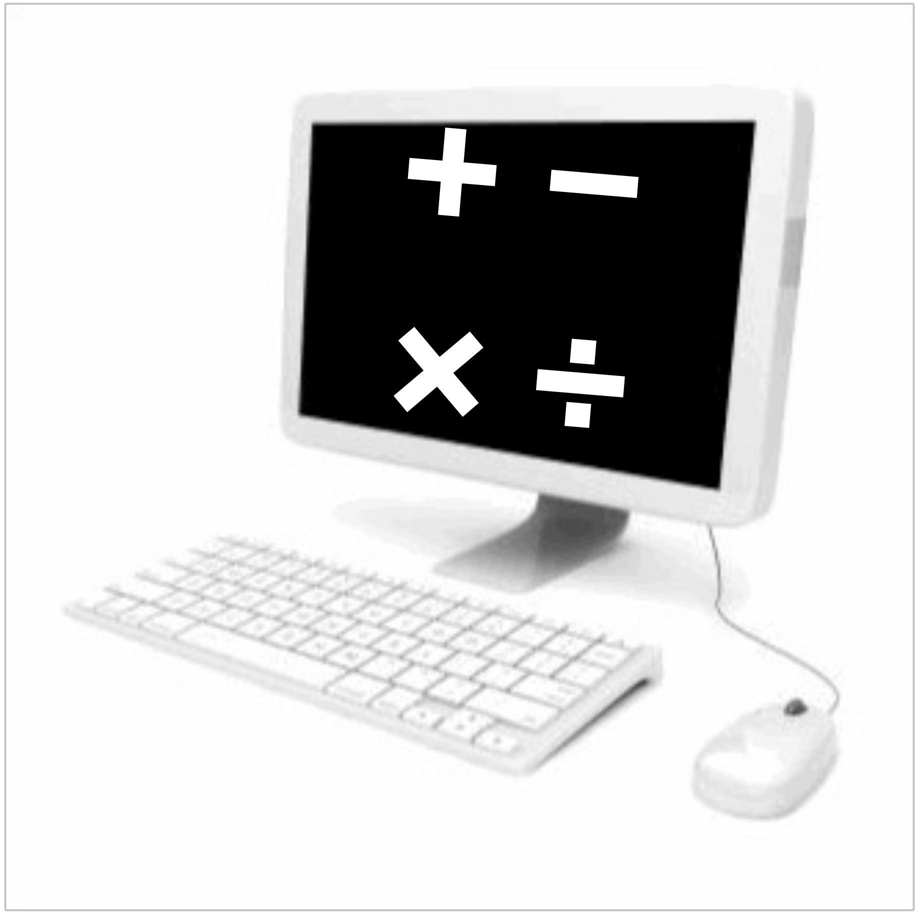
A computer.

It only works with numbers.

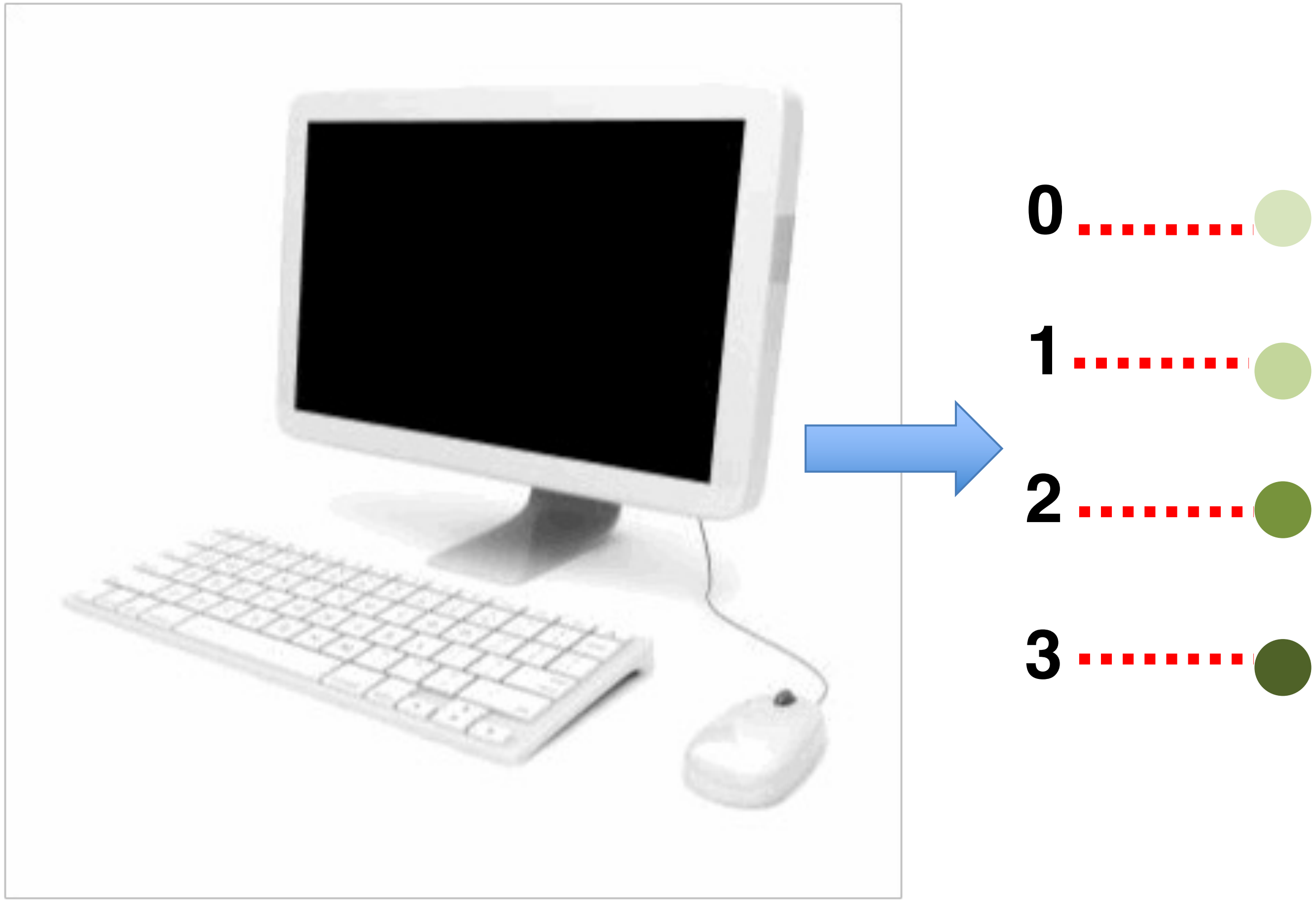
It cannot work with anything else.



Input needs to be ENCODED.



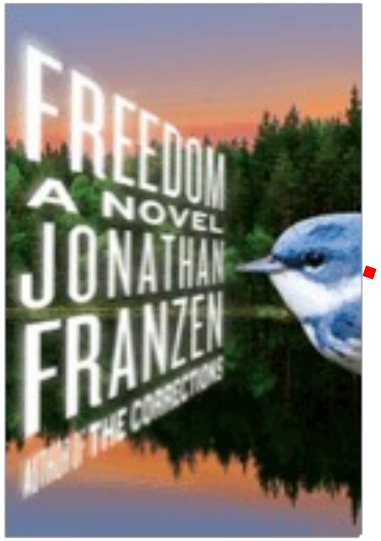
The computer works.



Output needs to be DECODED.

# COMPUTER SCIENCE

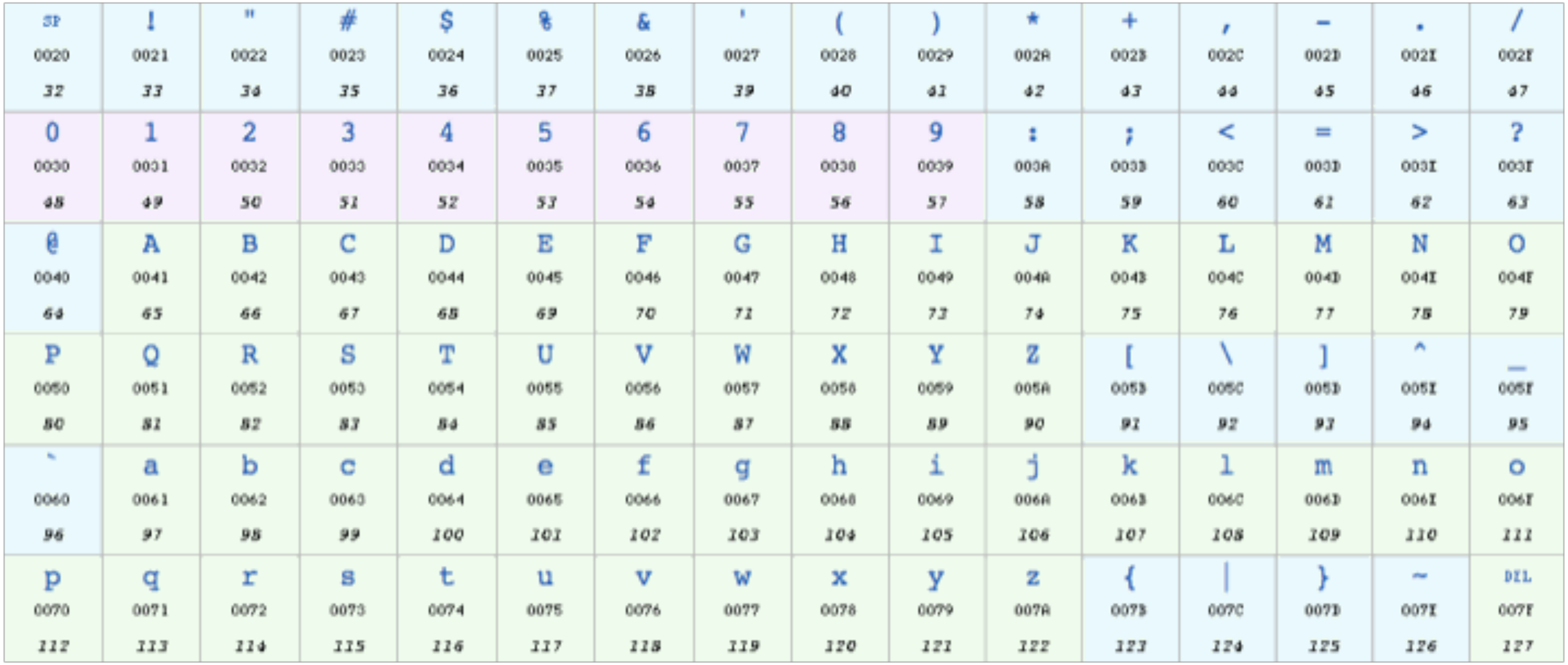
Let's see how we can create an encoding for texts.



8

The encoding of texts is based on the encoding of the characters that compose them: if I match each letter and each punctuation mark with a number, I obtain a one-to-one correspondence between letters and numbers. For this correspondence to be useful, it must be known and shared by all those who want to use a computer to exchange the texts thus encoded.

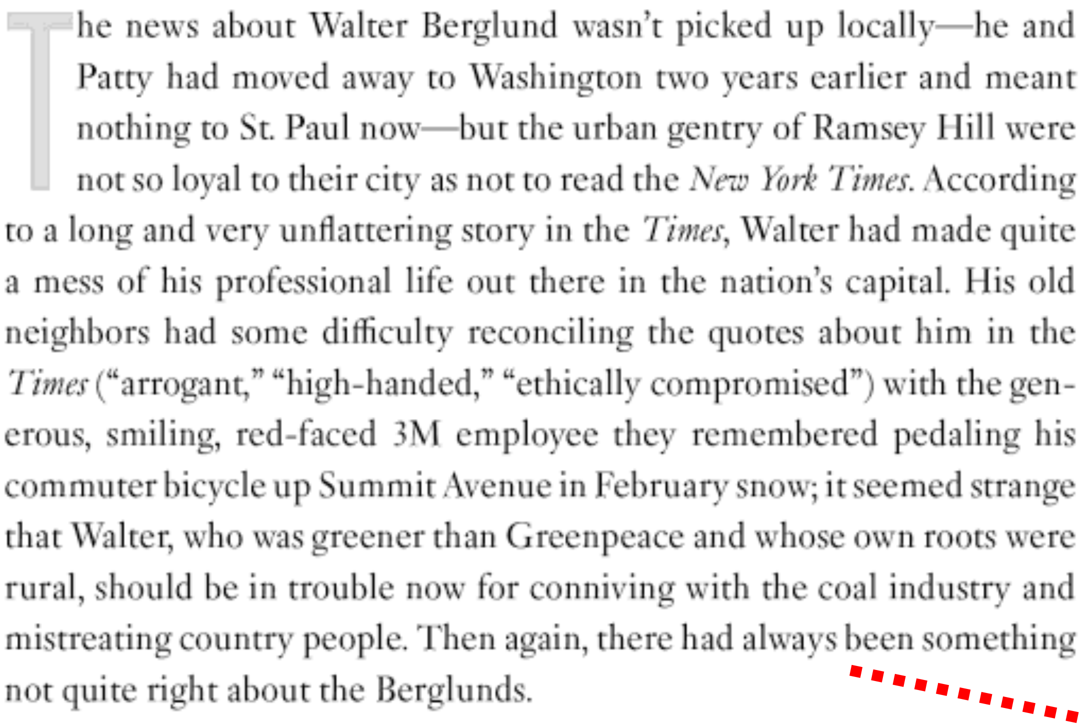
This table shows a well-known encoding: UTF-8



| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SP<br>0020<br>32 | !<br>0021<br>33 | "<br>0022<br>34 | #<br>0023<br>35 | $<br>0024<br>36 | %<br>0025<br>37 | &<br>0026<br>38 | '<br>0027<br>39 | (<br>0028<br>40 | )<br>0029<br>41 | *<br>002A<br>42 | +<br>002B<br>43 | ,<br>002C<br>44 | -<br>002D<br>45 | .<br>002E<br>46 | /<br>002F<br>47 |
| 0<br>0030<br>48 | 1<br>0031<br>49 | 2<br>0032<br>50 | 3<br>0033<br>51 | 4<br>0034<br>52 | 5<br>0035<br>53 | 6<br>0036<br>54 | 7<br>0037<br>55 | 8<br>0038<br>56 | 9<br>0039<br>57 | :<br>003A<br>58 | ;<br>003B<br>59 | <<br>003C<br>60 | =<br>003D<br>61 | ><br>003E<br>62 | ?<br>003F<br>63 |
| @<br>0040<br>64 | A<br>0041<br>65 | B<br>0042<br>66 | C<br>0043<br>67 | D<br>0044<br>68 | E<br>0045<br>69 | F<br>0046<br>70 | G<br>0047<br>71 | H<br>0048<br>72 | I<br>0049<br>73 | J<br>004A<br>74 | K<br>004B<br>75 | L<br>004C<br>76 | M<br>004D<br>77 | N<br>004E<br>78 | O<br>004F<br>79 |
| P<br>0050<br>80 | Q<br>0051<br>81 | R<br>0052<br>82 | S<br>0053<br>83 | T<br>0054<br>84 | U<br>0055<br>85 | V<br>0056<br>86 | W<br>0057<br>87 | X<br>0058<br>88 | Y<br>0059<br>89 | Z<br>005A<br>90 | [<br>005B<br>91 | \<br>005C<br>92 | ]<br>005D<br>93 | ^<br>005E<br>94 | _<br>005F<br>95 |
| `<br>0060<br>96 | a<br>0061<br>97 | b<br>0062<br>98 | c<br>0063<br>99 | d<br>0064<br>100 | e<br>0065<br>101 | f<br>0066<br>102 | g<br>0067<br>103 | h<br>0068<br>104 | i<br>0069<br>105 | j<br>006A<br>106 | k<br>006B<br>107 | l<br>006C<br>108 | m<br>006D<br>109 | n<br>006E<br>110 | o<br>006F<br>111 |
| p<br>0070<br>112 | q<br>0071<br>113 | r<br>0072<br>114 | s<br>0073<br>115 | t<br>0074<br>116 | u<br>0075<br>117 | v<br>0076<br>118 | w<br>0077<br>119 | x<br>0078<br>120 | y<br>0079<br>121 | z<br>007A<br>122 | {<br>007B<br>123 | \|<br>007C<br>124 | }<br>007D<br>125 | ~<br>007E<br>126 | DEL<br>007F<br>127 |

# UTF-8

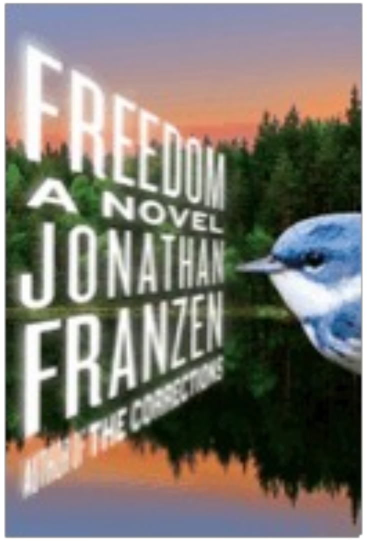
## Universal Character Set Transformation Format – 8bit



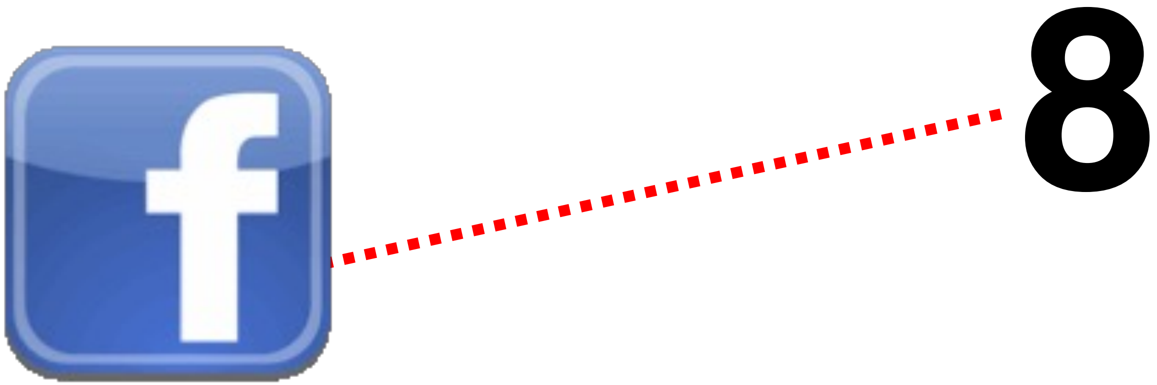
The news about Walter Berglund wasn't picked up locally—he and Patty had moved away to Washington two years earlier and meant nothing to St. Paul now—but the urban gentry of Ramsey Hill were not so loyal to their city as not to read the *New York Times*. According to a long and very unflattering story in the *Times*, Walter had made quite a mess of his professional life out there in the nation's capital. His old neighbors had some difficulty reconciling the quotes about him in the *Times* ("arrogant," "high-handed," "ethically compromised") with the generous, smiling, red-faced 3M employee they remembered pedaling his commuter bicycle up Summit Avenue in February snow; it seemed strange that Walter, who was greener than Greenpeace and whose own roots were rural, should be in trouble now for conniving with the coal industry and mistreating country people. Then again, there had always been something not quite right about the Berglunds.

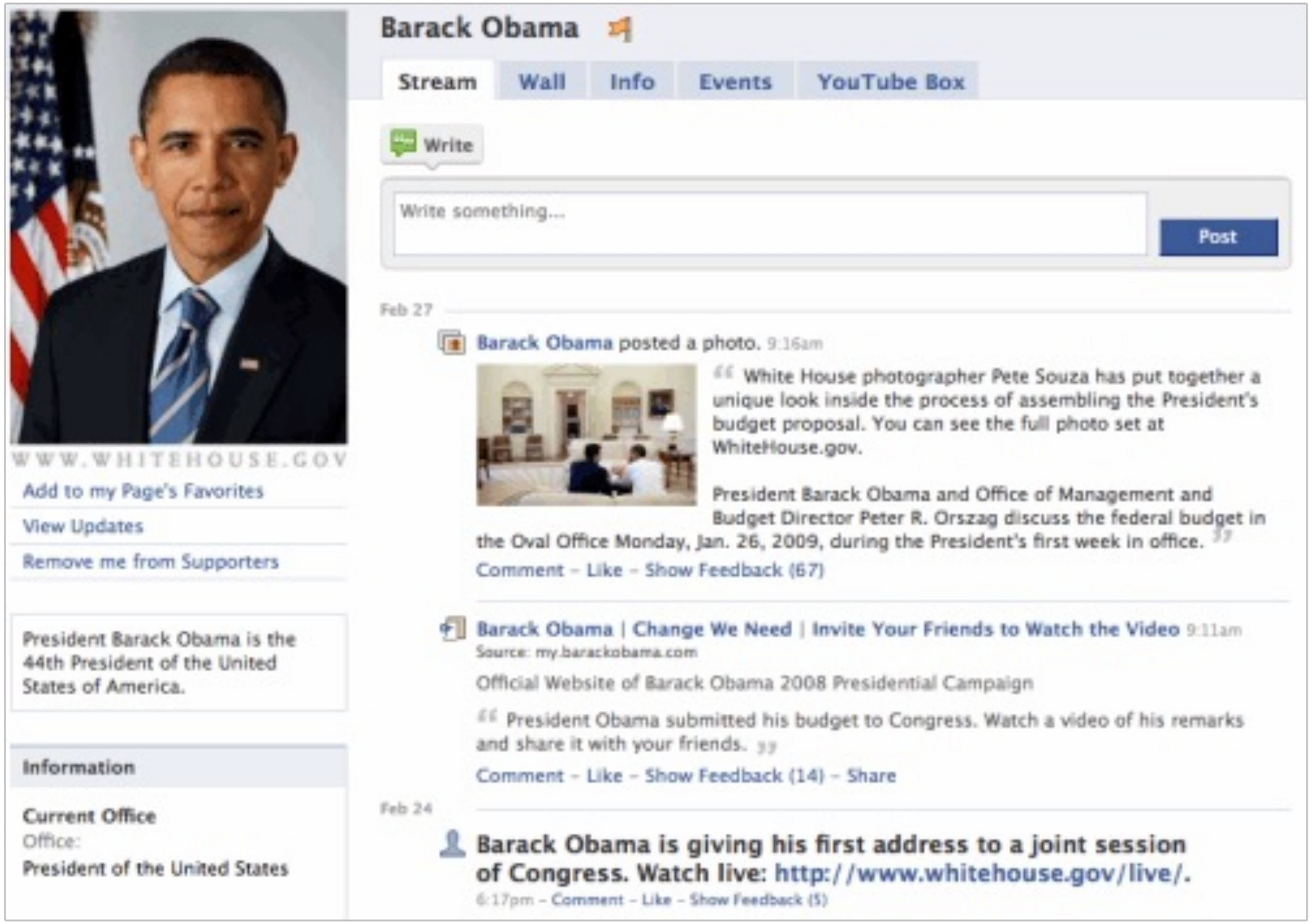
In order to be processed by a computer, the words of Jonathan Franzen's novel "Freedom" must be encoded in the form of a sequence of numbers. Each number corresponds uniquely to a character in the text.

```
108927350456345347653845098753094857039
```



Let's see how social networks like Facebook can also be encoded.





President Barack Obama is the 44th President of the United States of America.

Information

Current Office
Office:
President of the United States

Feb 27

Barack Obama posted a photo. 9:16am

“White House photographer Pete Souza has put together a unique look inside the process of assembling the President's budget proposal. You can see the full photo set at WhiteHouse.gov.

President Barack Obama and Office of Management and Budget Director Peter R. Orszag discuss the federal budget in the Oval Office Monday, Jan. 26, 2009, during the President's first week in office.”

Comment - Like - Show Feedback (67)

Barack Obama | Change We Need | Invite Your Friends to Watch the Video 9:11am
Source: my.barackobama.com

Official Website of Barack Obama 2008 Presidential Campaign

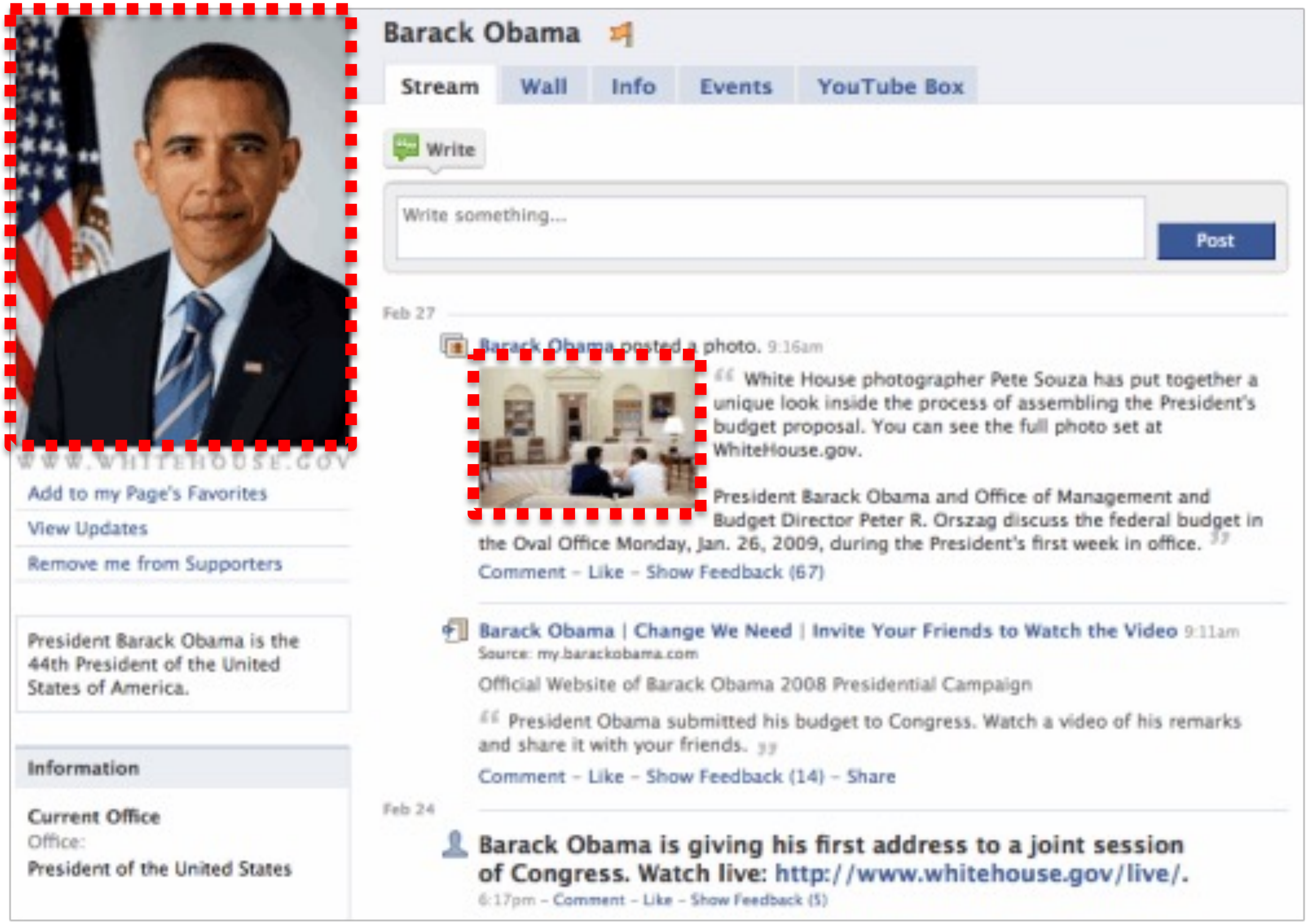
“President Obama submitted his budget to Congress. Watch a video of his remarks and share it with your friends.”

Comment - Like - Show Feedback (14) - Share

Feb 24

**Barack Obama is giving his first address to a joint session of Congress. Watch live: http://www.whitehouse.gov/live/.**
6:17pm - Comment - Like - Show Feedback (5)

What we see on our computer of a social network is nothing more than a set of texts and images.



Barack Obama

Add to my Page's Favorites

View Updates

Remove me from Supporters

President Barack Obama is the 44th President of the United States of America.

Information

Current Office

Office:

President of the United States

Feb 27

Barack Obama posted a photo. 9:16am

"White House photographer Pete Souza has put together a unique look inside the process of assembling the President's budget proposal. You can see the full photo set at WhiteHouse.gov.

President Barack Obama and Office of Management and Budget Director Peter R. Orszag discuss the federal budget in the Oval Office Monday, Jan. 26, 2009, during the President's first week in office."

Comment - Like - Show Feedback (67)

Barack Obama | Change We Need | Invite Your Friends to Watch the Video 9:11am

Source: my.barackobama.com

Official Website of Barack Obama 2008 Presidential Campaign

"President Obama submitted his budget to Congress. Watch a video of his remarks and share it with your friends."

Comment - Like - Show Feedback (14) - Share

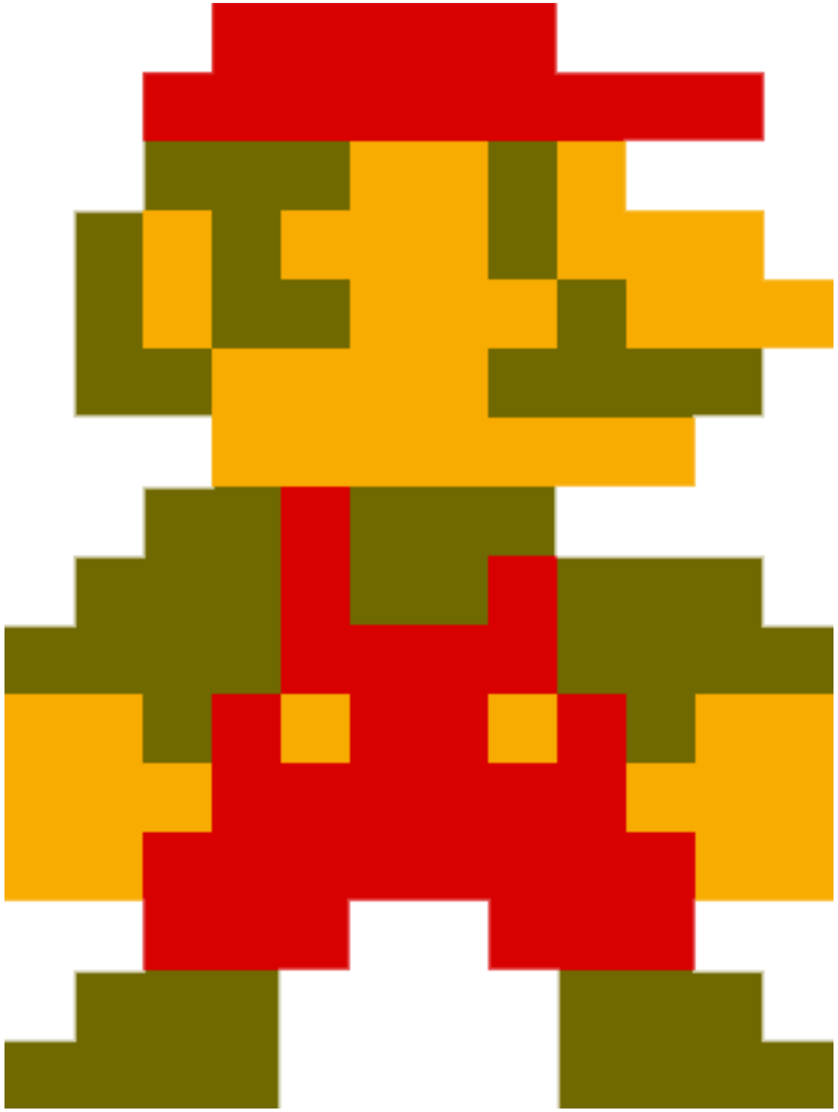
Feb 24

**Barack Obama is giving his first address to a joint session of Congress. Watch live: http://www.whitehouse.gov/live/.**

6:17pm - Comment - Like - Show Feedback (5)

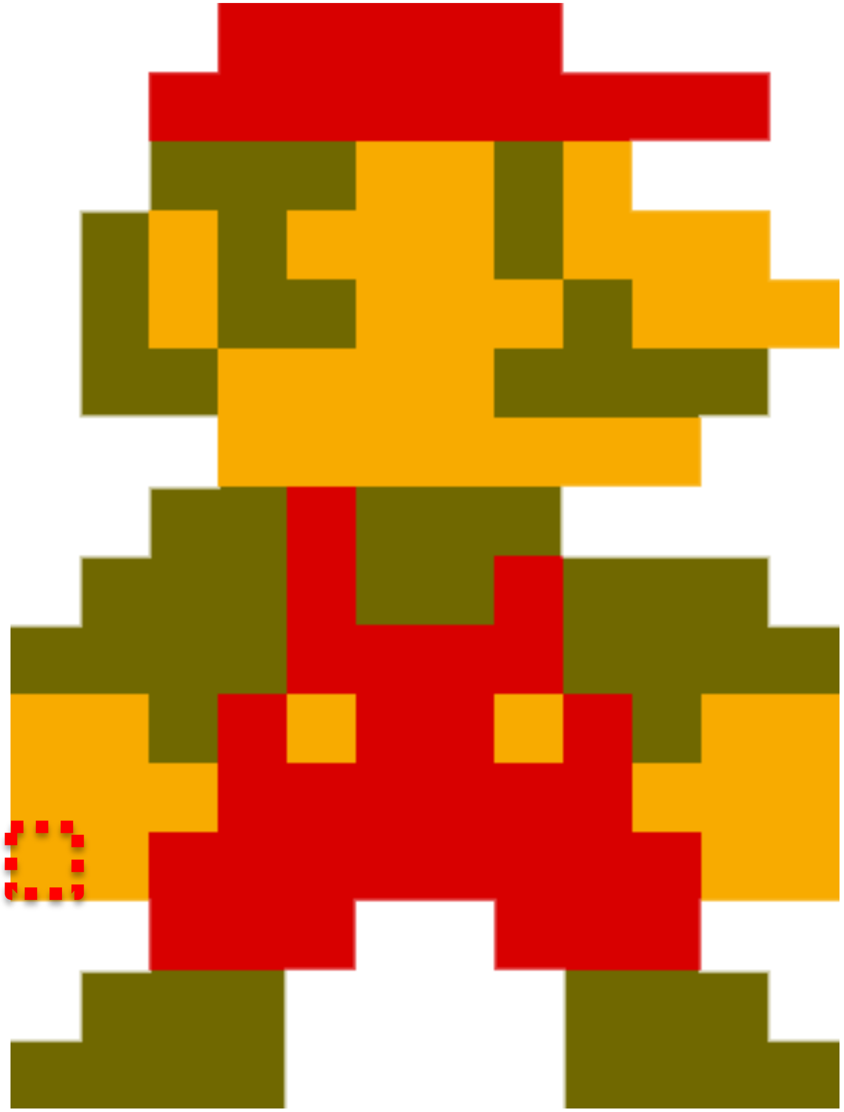
We've already dealt with texts, so now let's focus our attention on images.

Looking at a very detailed photo of sweets from Bergamo, it is difficult to think that there could be a correspondence with the natural numbers.



By looking at a low-resolution image of Mario, you can instead understand how a photo can be encoded.

# pixel: picture element

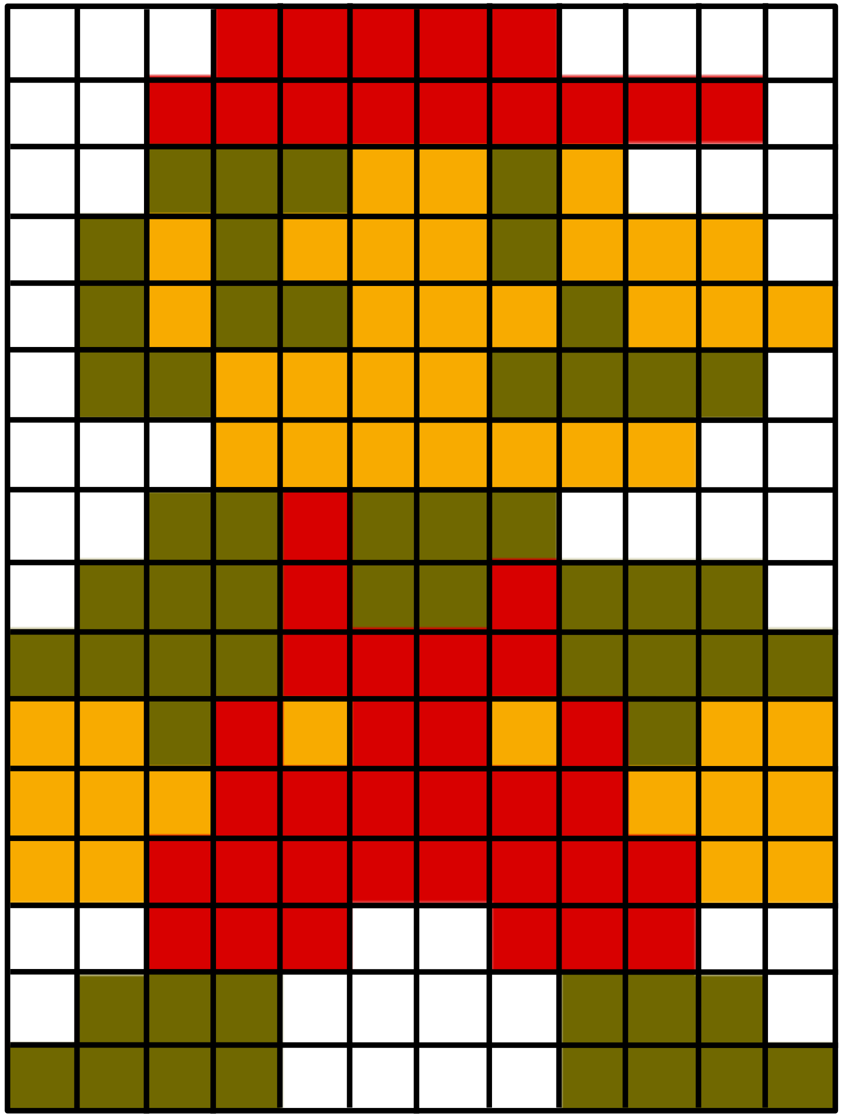


In the image of Mario, you can recognize basic square elements, characterized by the positions they occupy, and by their color: they are pixels.

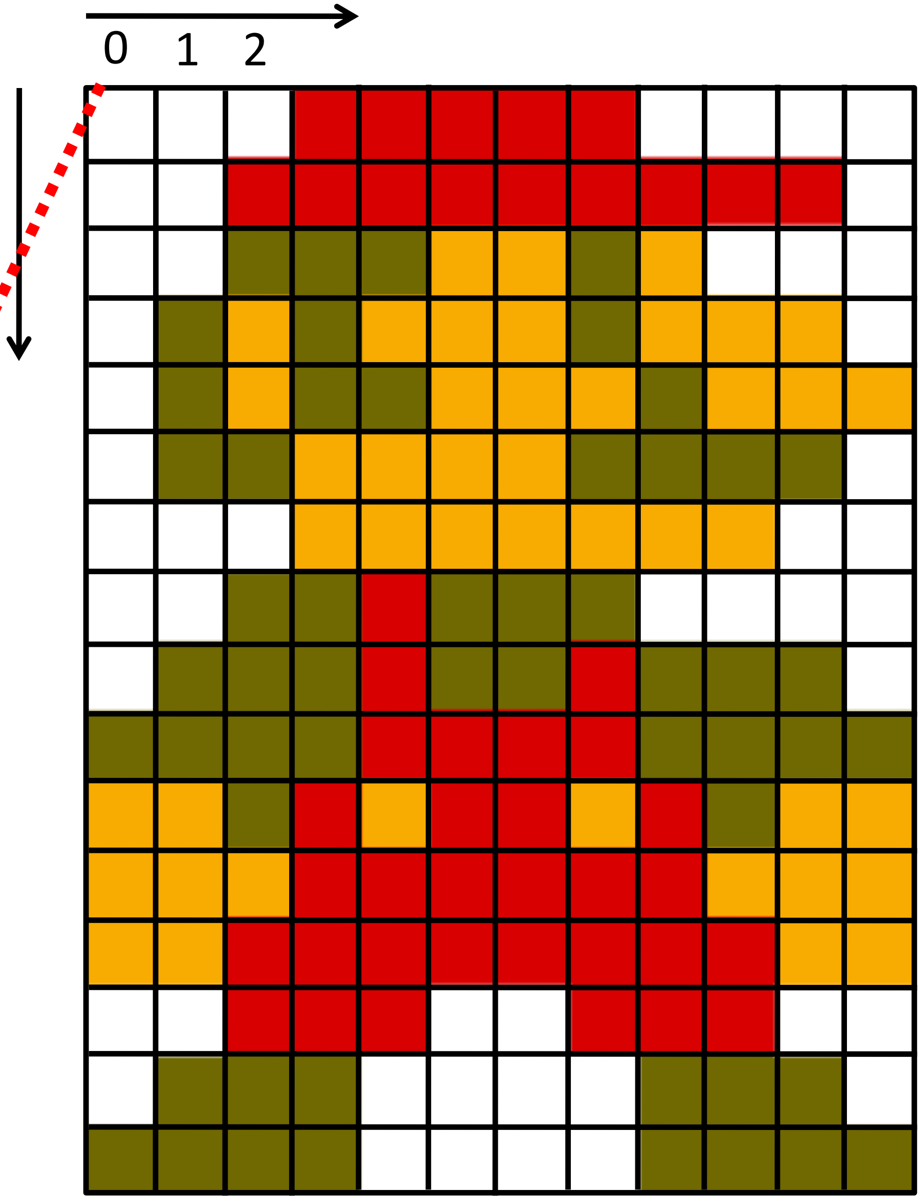
This photo is also made of pixels. We just don't see them because they are too small. In this image there are 12 million of them.

**pixel (1/12000000)**

Image encoding is based on the pixels that compose it. In the same way as text is considered as a sequence of characters to be encoded…

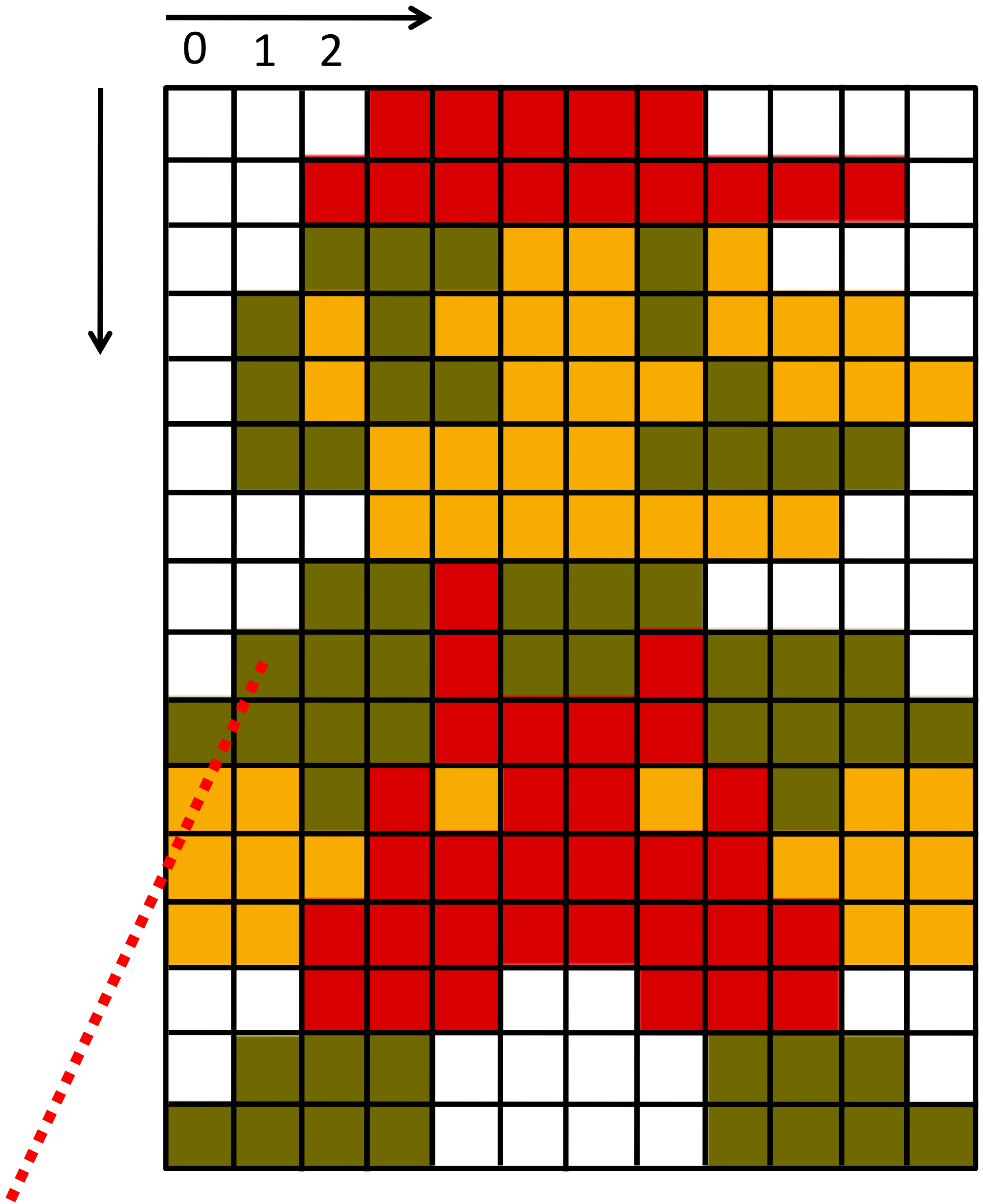


…so we consider a photo a sequence of pixels. If we can encode one pixel, we can encode an entire image.

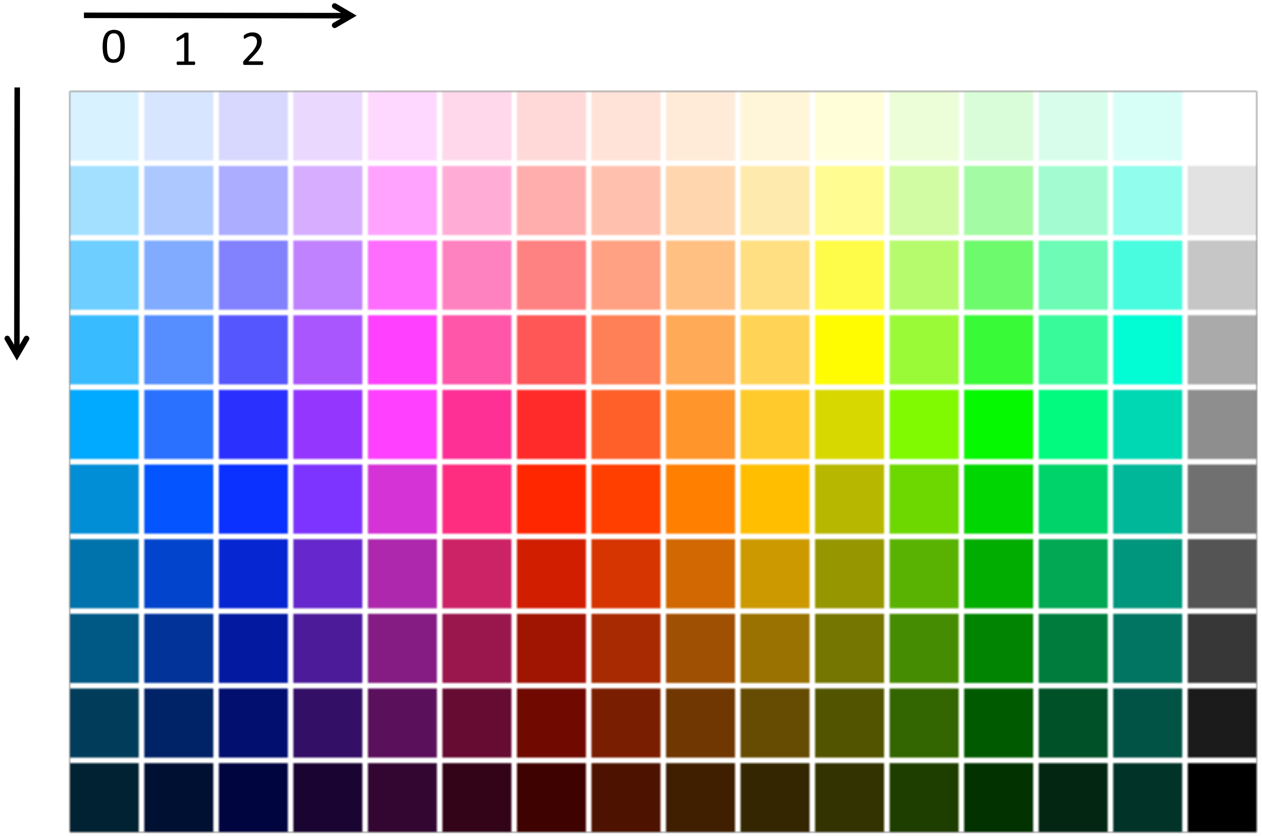


**encoding of the pixel's position**

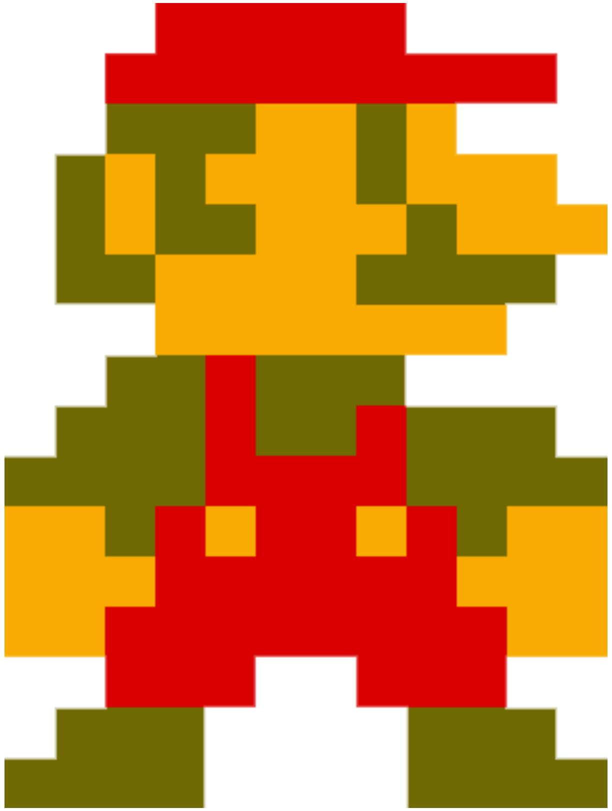
We said that the first characteristic of a pixel is its position. If we imagine inserting the image in a Cartesian plane, we are able to identify the position of the pixel via its numerical coordinates



**what about its color?**



If we imagine organizing colors in a table, in the same way as pixels we are able to associate a numerical value with each color, corresponding to the coordinates of the color in the table.

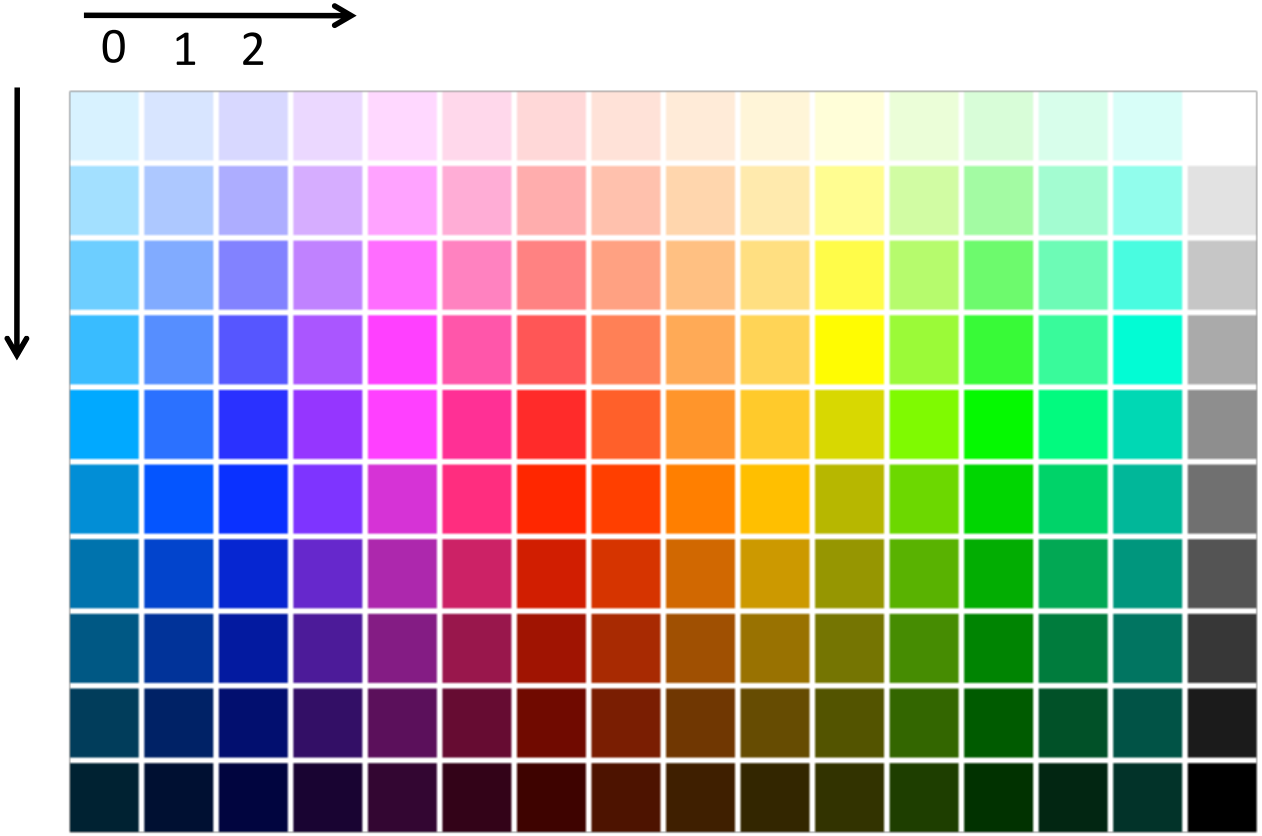


The pixels that make up Mario, therefore, can be coded based on their position and their color, allowing the image, a sequence of pixels, to be expressed in the form of a sequence of numbers.

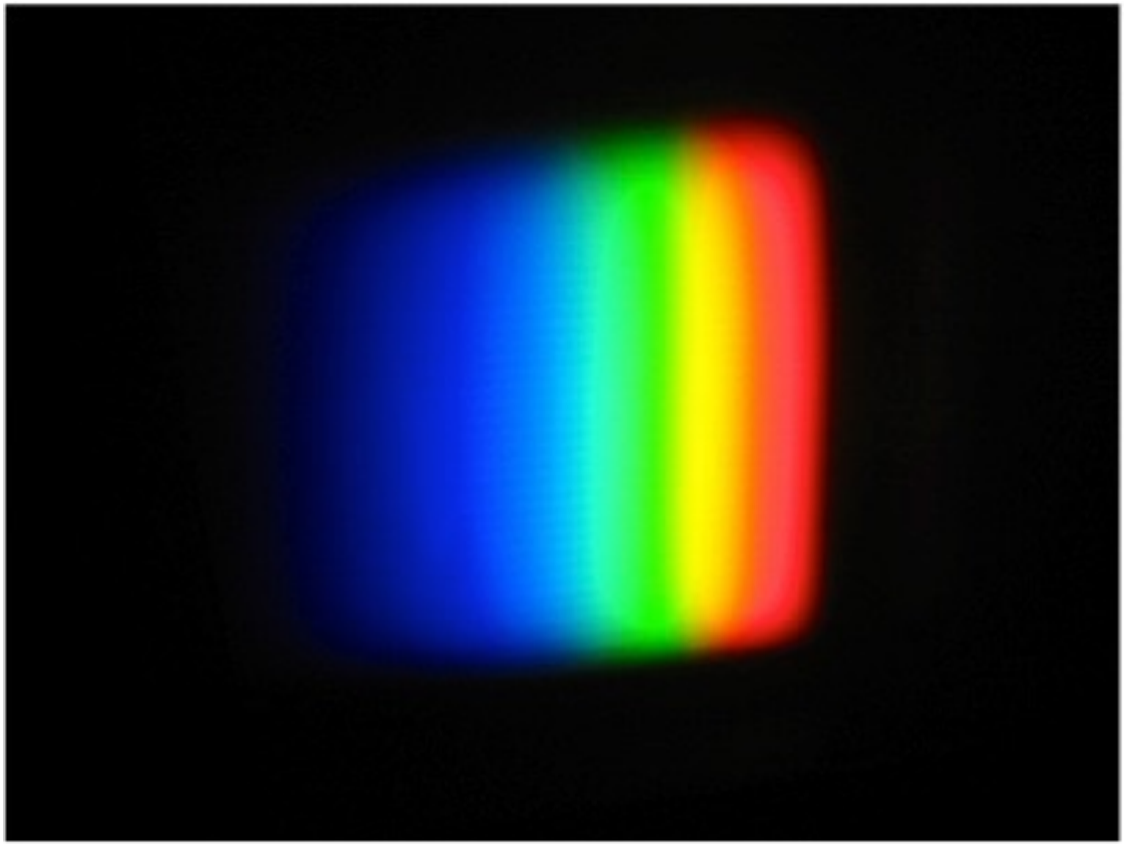
```
567872872983093948748974987498748467101
187627862876783546354137676129123621352
413651243873614983502385746754779090939
423847293847283478237492384723894728347
2983742836471651555625465463546354376
473658475984759824757671511019824928493
849340283918091740375474783474736478364
734637843940109134813409463074560384756
038476501837465087314650138746507834560
384756037486578346574658734506183745601
837465018376456758403876573480187364571
088573465783104587134653178451103874650
```

We proceed in a similar way for the 12-million-pixel photo. It is no coincidence that more detailed photos take up more space in a computer memory: there are simply more encoded pixels.

```
1089273504563453476538450987530948570393
8456287650394175389457345893490583409558
4478262965673463746374633493489898101001
0294938483748734873948734817617647394887
3984739487384718937491837483748374837443
8147183974913847918347813471398471839447
3184781347183473874834783473847387487338
4734837487348738473847384783478347384778
3478347918374981374834831387465078345600
3847560374865783465746587345061837456001
8374650183764567584038765734801873645711
0885734657831045871346531784511038746500
```



As you will have noticed, in the table that allows color coding, not all shades are present. Just take two adjacent squares and imagine taking the intermediate color: it is not clearly present in the table, so it is not an encoded color.



This means that only a small part of the colors in the spectrum can be encoded. Numerous nuances (infinite, to be precise) remain left out of the coding. On the other hand, it is unthinkable to have a table of infinite dimensions to accommodate the infinite shades of colors.

This is the problem of the conversion from physical phenomena to “digital” (i.e. made of digits) descriptions: the encoding of physical phenomena always involves a loss of information. Drawing a comparison with mathematics, physical phenomena are characterized by infinitesimal nuances similar to real numbers, while encodings are correspondences with natural numbers, which, despite being infinite, are still much smaller than real numbers and cannot act as references to the aforementioned nuances.

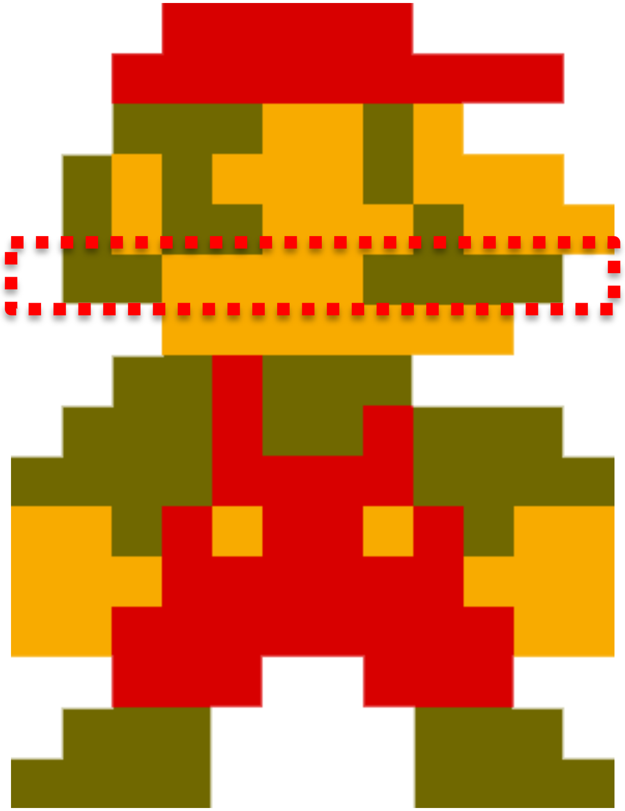
# Physical [Analog]
# vs.
# Digital

It is like when people say that looking at a photo can never equal seeing the real thing. In reality, three-dimensionality aside, current color encoding can capture millions of shades, many of which are indistinguishable to the human eye. Therefore, albeit with losses, encoding allows us to have more than good approximations of reality.

We mentioned before that the detailed photo of the sweets results in a much longer encoded sequence of numbers than the very simple drawing of Mario. The length of the encodings naturally depend on what is encoded: the greater the information to be encoded, the longer the result, and in fact the photo of the sweets (with all the details and nuances given by the 12 million pixels) certainly takes up more space in the computer than the Mario drawing (which, in fact, is a sequence of 16 x 12 pixels).

# Compression

There are, however, different ways of describing the sequence of encoded pixels, and, if we choose wisely, we can obtain a shorter description of the same sequence, so that it takes up less space on the computer, and also saves time in transmitting it from one computer to another. We call "compression" a way to obtain sequences of symbols that describe a particular piece of information in a shorter way than the simple list of encodings of each element that composes it.



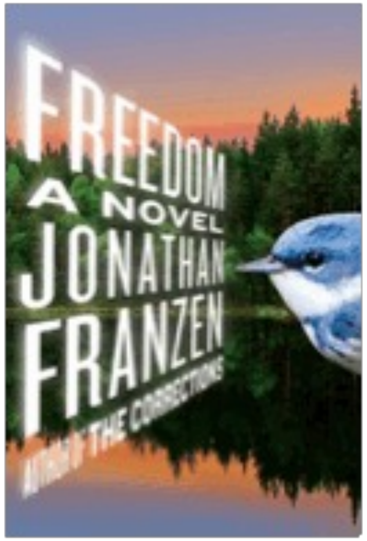
**white pixel, green pixel, green pixel, yellow pixel, yellow pixel, yellow pixel, yellow pixel, green pixel, green pixel, green pixel, green pixel, white pixel**

→

**1 white pixel,**
**2 green pixels,**
**4 yellow pixels,**
**4 green pixels,**
**1 white pixel**

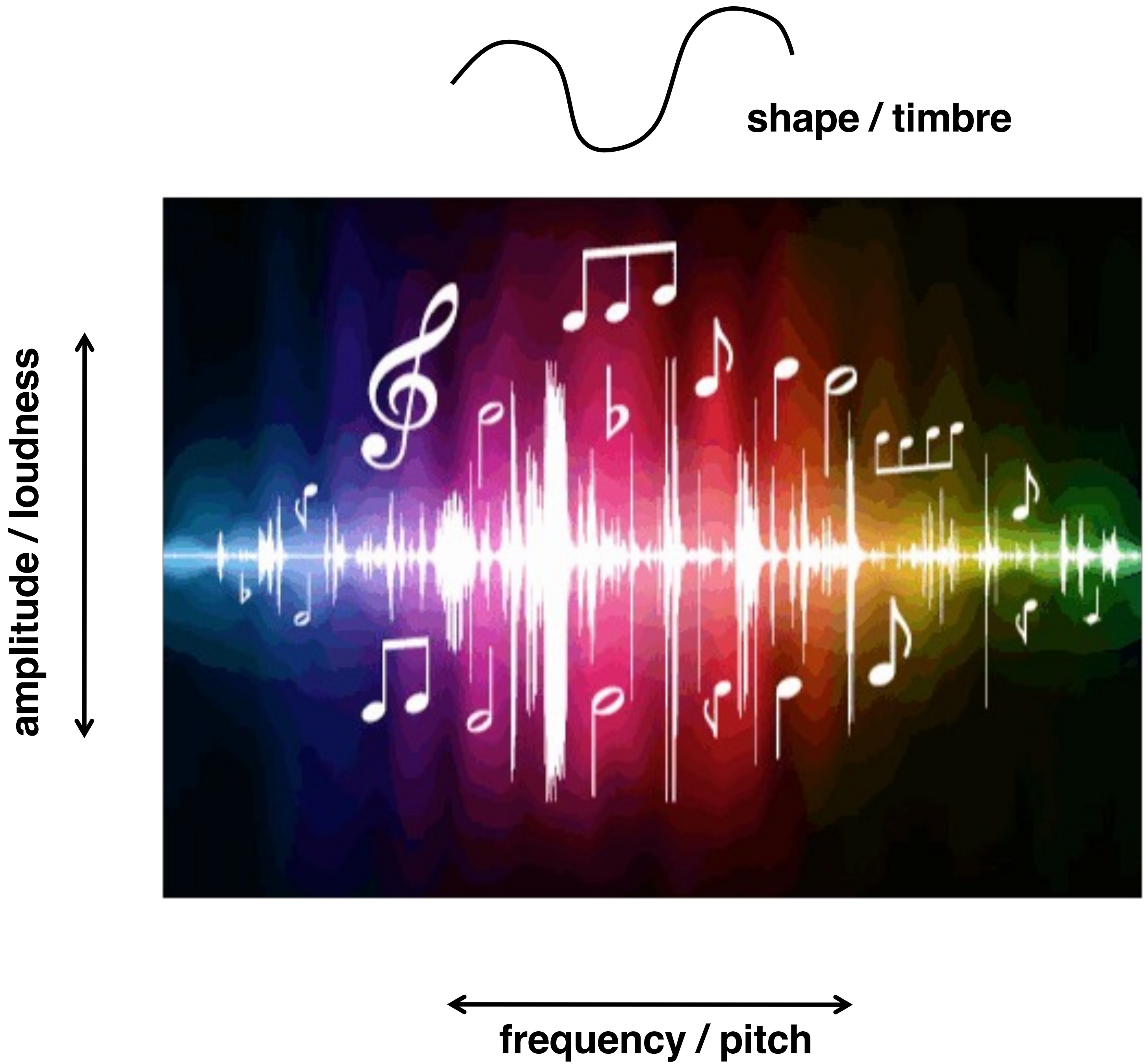
The description of the Mario image, for example, can be compressed by describing its pixels as shown in the second red box.

Any type of information that can be encoded can be compressed. The compression ratio, i.e. how much we can summarize the description, depends on the content of the information. (Imagine having to describe an all-white 16 x 12 rectangle. How would you do it?)



Let's now see how encoding allows you to have music on your computer.

**8**



shape / timbre

amplitude / loudness

frequency / pitch

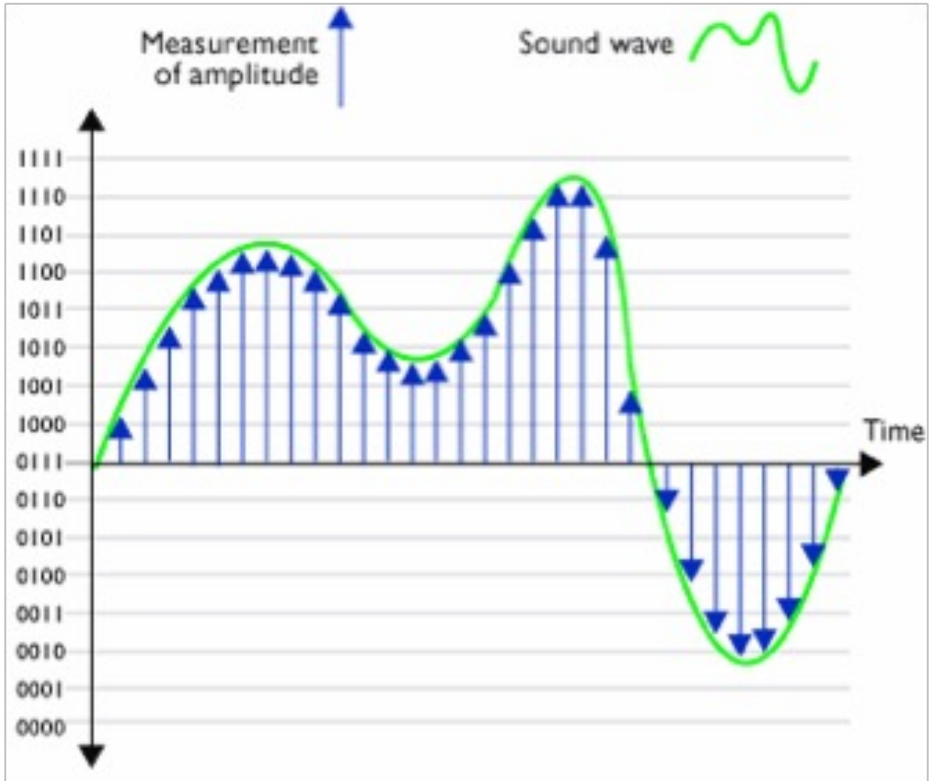
Sounds, and therefore also music, are due to vibrations of the air (or of any medium*) which propagate in the form of waves and which, hitting our eardrums, give rise to what we perceive with our ears. Sound waves are characterized by an amplitude (which determines the loudness of the sound: the wider the wave, the louder the sound), a frequency (which determines the pitch of the note: high notes correspond to high frequencies), and by a shape (which determines the timbre of the sound: my voice, your voice, piano, violin, cymbals, a doorbell, …)

*sounds do not propagate in a vacuum

# Sampling

The encoding of sounds is based on the encoding of the waves that produce them, which is in turn based on a procedure called sampling.

Sampling consists in considering the wave that constitutes the sound only in certain moments of time. Imagine describing the sound wave (with its amplitude, frequency, and shape) in a Cartesian plane, and considering only certain points of this curve.



Each measurement is assigned a number (byte) according to its amplitude. The end result is a file comprising a string of bytes, eg ...
1001 1110 0001 1010 0111 0100 1111 1101 etc

These points correspond, in the Cartesian plane, to precise coordinates, whose numerical value is used as coding of the “sampled” sound. The encoded sequence of the samples constitutes the encoding of the entire sound data represented by the wave.

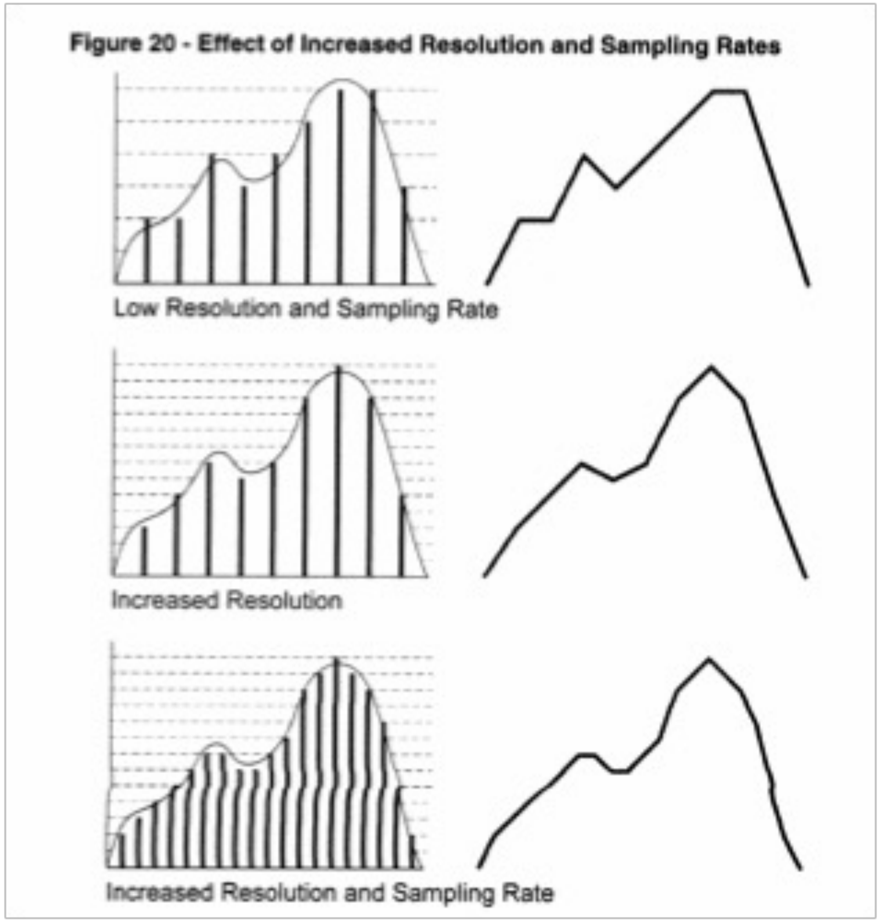


Figure 20 - Effect of Increased Resolution and Sampling Rates

Low Resolution and Sampling Rate
Increased Resolution
Increased Resolution and Sampling Rate

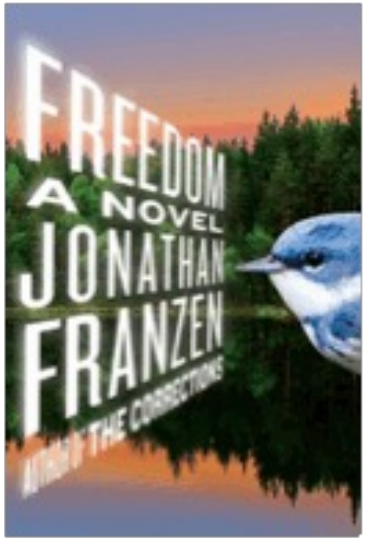
**sampling frequency**

There is no universal criterion for establishing the distance between one sample and the next (or its inverse, known as “sampling frequency”). It is easy to imagine that samples that are closer to each other (higher sampling frequency) correspond to a longer encoding, and also a reconstruction that is more faithful to the original wave. Even in the case of sound and music encodings, there are compression methods to synthesize their description. The famous MP3 files are called this way because they take their name from a specific sound encoding technique with compression. Even in this context there are people who say that the sound of an MP3 on a computer will never equal the quality of a live concert: it’s all a question of approximations.

Encoding a sound is used to make it processable by a computer (for example, to allow the transfer of a song from the iTunes online store to our computer). However, even in the digital age, we continue to listen with our ears, and to make our listening possible, we need sound waves that propagate in the air. A reconversion from numerical coding to sound waves is therefore necessary: that is, we need speakers which, controlled by the electrical signals produced by the computer according to the numbers contained in the coding of the song, make membranes vibrate which produce waves that we perceive as sounds and music.

**again, physical vs digital**

Finally, let's see how to proceed with video encoding.



8

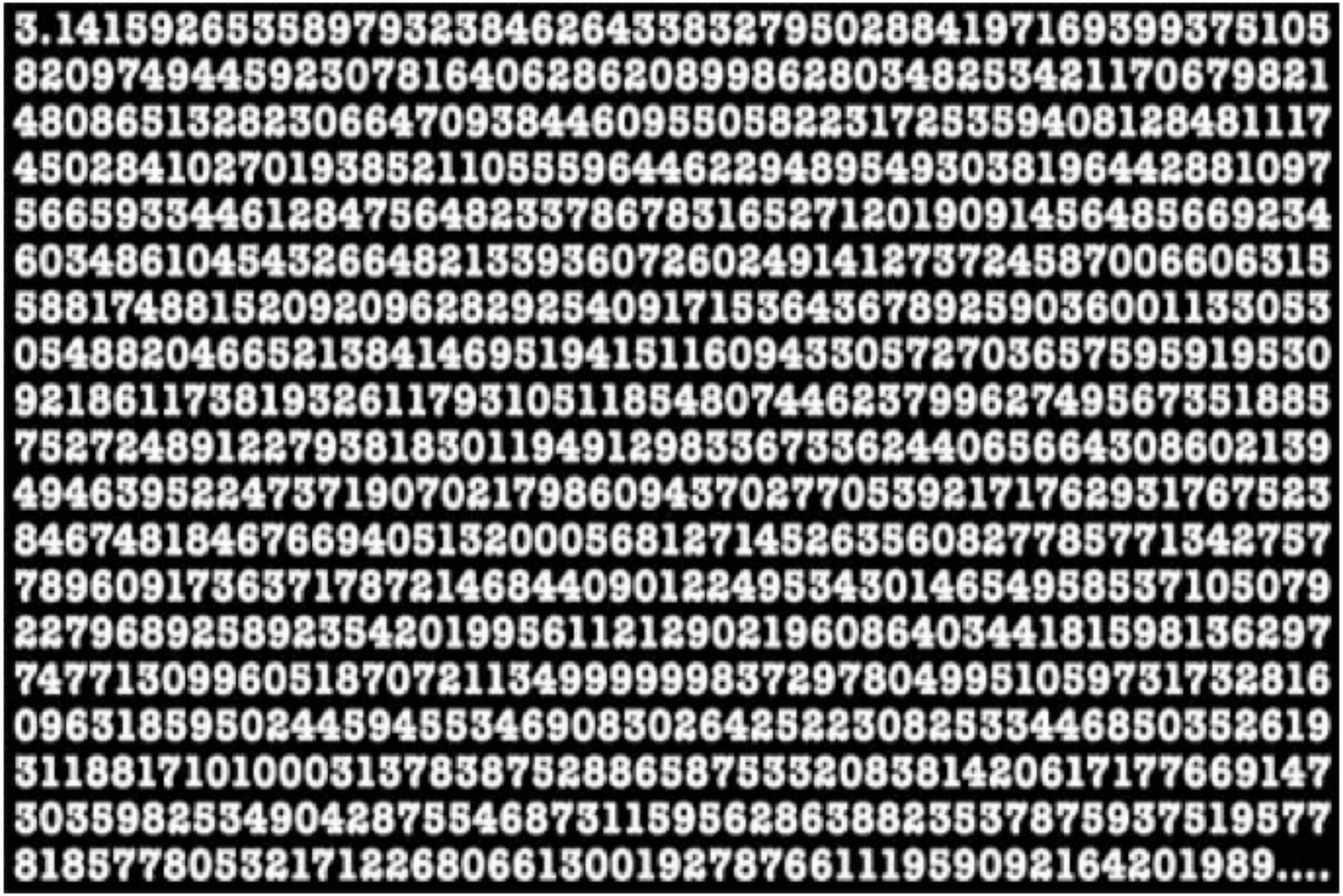
Actually, having image and sound encoding techniques available, it is easy to imagine that they can be combined to create video encoding.

Additional techniques are needed to take into account the synchronization between images and sounds, and compression techniques based on the idea of not describing all the pixels of each image, but only describing the initial one and then focusing on the differences between an image and the next (better compression with very similar frames, worse compression with scene changes).

=

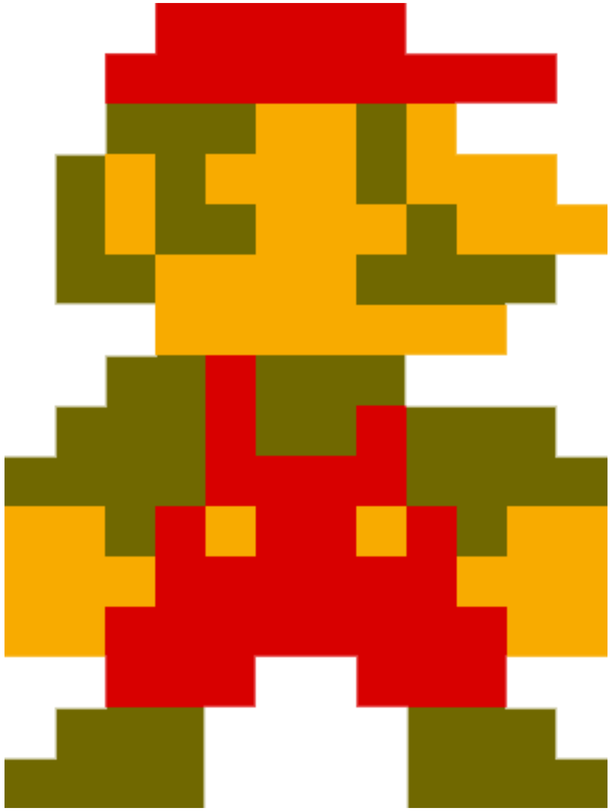
+

+ . . .



3.141592653589793238462643383279502884197169399375105
8209749445923078164062862089986280348253421170679821
48086513282306647093844609550582231725359408128481117
45028410270193852110555964462294895493038196442881097
56659334461284756482337867831652712019091456485669234
60348610454326648213393607260249141273724587006606315
58817488152092096282925409171536436789259036001133053
05488204665213841469519415116094330572703657595919530
92186117381932611793105118548074462379962749567351885
75272489122793818301194912983367336244065664308602139
49463952247371907021798609437027705392171762931767523
84674818467669405132000568127145263560827785771342757
78960917363717872146844090122495343014654958537105079
22796892589235420199561121290219608640344181598136297
74771309960518707211349999998372978049951059731732816
0963185950244594553469083026425223082533446850352619
31188171010003137838752886587533208381420617177669147
3035982534904287554687311595628638823537875937519577
818577805321712268066130019278766111959092164201989....

As already mentioned, the approximation that occurs when passing from physical phenomena such as colors and sound waves to numerical encoding suitable for a computer is reminiscent of the relationship between real numbers and natural numbers in mathematics. Perhaps the use of natural numbers to talk about the concept of encoding may seem limiting and may seem to invite the involvement of rational numbers, but the underlying argument does not change: even with rational numbers, the encodings are unable to faithfully reproduce a physical phenomenon, but they bring with them inevitable approximations. The number in the previous slide, for example, represents only a part of the decimal expansion of the number π . We can approximate it as we like, but part of the description of π will still be left out and cannot be included, since no calculator is capable of containing an infinite sequence of digits. The important thing is that the approximation reaches such detail that the reconstruction of the physical phenomenon is indistinguishable from the original to the eyes (or ears) of the human being who uses the computer to process it.

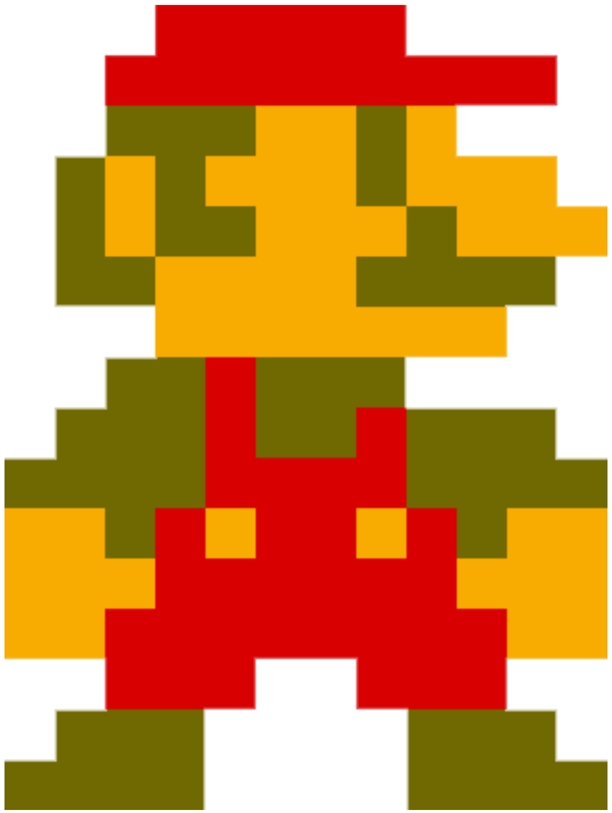


Let’s consider again the concept of encoding with the example of the image of Mario. To be processed by a computer, it must be transformed into a sequence of digits.

```
567872872983093948748974987498748467101
187627862876783546354137676129123621352
413651243873614983502385746754779090939
423847293847283478237492384723894728347
29837428364716515515625465463546354376
473658475984759824757671511019824928493
849340283918091740375474783474736478364
734637843940109134813409463074560384756
038476501837465087314650138746507834560
384756037486578346574658734506183745601
837465018376456758403876573480187364571
088573465783104587134653178451103874650
```

Actually, these digits must themselves be encoded to be processed by a computer.

This further encoding transforms them into a sequence of 0s and 1s.



```
01010101010101000100111010101010101010101
01110101000010101010101011101010101010110
10101001001111111010101010101010010101
01010101011111100101000110101010100011101
010101011110101101010001010100111010011
01010011100010101110101010001010101011010
10001101101010111010100101000101010101000
10111010101010111001010101010100010101010
101010101110101010100010101010111010101
010110101011111010111000001110001110011
10011010101110011100011100010101010101111
00001110000101010101011001010101011111000
```

# What is a bit?

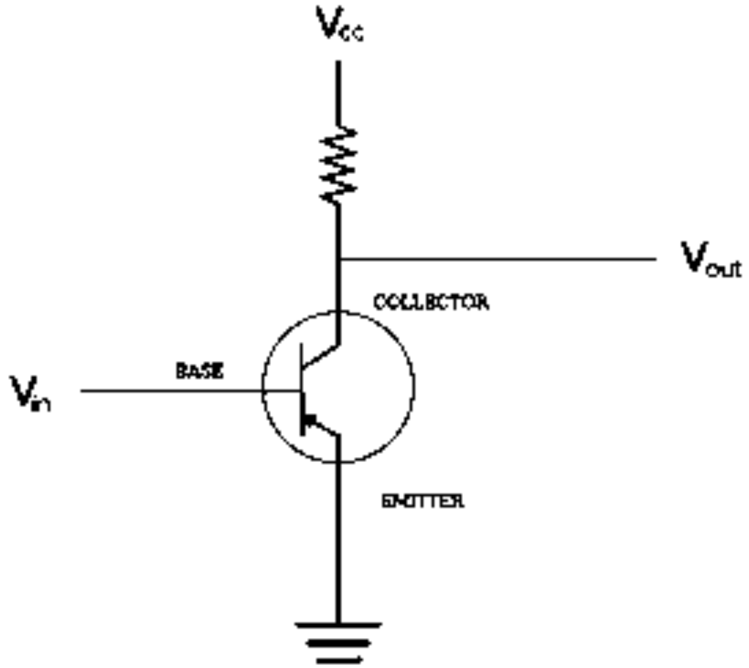
It is the basic data unit processed by a computer: it is 0 or 1.

bit = binary + digit.

8 bit (b) = 1 Byte (B).

The encoding that uses only 0 and 1 is called binary encoding (because it only uses 2 digits).

# Why 0 and 1?



**High electric voltage: 1**
**Low electric voltage: 0**

The electronic circuits that make up a computer are built to respond to high or low voltage electrical signals. High voltage is interpreted as a “1”, while low voltage is interpreted as a “0”. The restriction to only two values has no physical reason: it is possible to build circuits that respond with numerous voltage levels. For example, we could have 10 different ones, interpretable as the 10 digits (from “0” to “9”) that are usually used in mathematics.

However, the advantages of having only two signals are numerous:
the circuits are simpler to make and cost less; furthermore, the output signals, even in the presence of disturbances due to natural causes, are easier to interpret, with less possibility of error. (e.g. $0.8 \approx 1$)

# Multiples

1 KB (kilo) = 1000 B

1 MB (mega) = 1 million B

1 GB (giga) = 1 billion B

1 TB (tera) = 1000 billion B

# Binary encoding of numbers

**Numbers with base 10:**

$$215 = 2\times10^2 + 1\times10^1 + 5\times10^0$$

**Numbers with base 2:**

$$110010111 = 1\times2^8 + 1\times2^7 + 0\times2^6 + 0\times2^5 + 1\times2^4 + 0\times2^3 + 1\times2^2 + 1\times2^1 + 1\times2^0$$

## Exercise

**Convert the following numbers from base 2 to base 10: 101, 1000, 11011.**

## Exercise

Convert the following numbers from base 10 to base 2: 8, 23, 144, 201.

**While the definition of a binary system helps us solve the first exercise, we must find new methods to solve the second. There are two.**

**First method:**

**given the number n in base 10, we look for the largest power of 2 that is less than or equal to n.**

**If it is the same, we have solved the problem: we write a 1 in the position corresponding to that power of 2, followed by zeros.**

**For example, 8 is a power of 2: $2^3$ to be precise, so its binary encoding will be 1000.**

**However, if the largest power of 2 that is less than or equal to n is less than n (let's call it k), let's set it aside and we calculate the difference n-k.**

**We repeat the same procedure with n-k, and look for the largest power of 2 that is less than or equal to it.**

**We continue until we are able to express n as a sum of powers of 2.**

**We take the list of powers and write a 1 in the corresponding positions, 0 in the others.**

**For example, the largest power of 2 contained in 23 is 16 ($2^4$). Their difference is 7, in which 4 ($2^2$) is contained. The difference is 3 where there is 2 ($2^1$), after which only 1 ($2^0$) remains.**

**Writing the powers of 2 present in order we get 10111.**

**Second method:**

**we divide the number by 2 and obtain quotient and remainder.**

**As long as the quotient is not 0, we take it as the new dividend and continue dividing. When we get zero quotient, we have to write the list of remainders in reverse order to get the binary encoding of the initial number.**

**Let's take 144 as an example:**

**144 : 2 = 72 with remainder 0**
**72 : 2 = 36 with remainder 0**
**36 : 2 = 18 with remainder 0**
**18 : 2 = 9 with remainder 0**
**9 : 2 = 4 with remainder 1**
**4 : 2 = 2 with remainder 0**
**2 : 2 = 1 with remainder 0**
**1 : 2 = 0 with remainder 1**

**144 in base 2 is 10010000 (these bits are the remainders in the reverse order of writing)**