university
of
tyumen

school
of advanced
studies

Information Technology - basic

Lecture 2

Logical representation of numbers

Fabio Grazioso - *April 2018*

Todays' lecture

Computer problem-solving

Summary of the lecture

- Data representation
- Numbering systems
  - binary
  - decimal
  - hexadecimal
- Text encodings

numbering systems

Fox p57

“normal” numbers

234

3980

637821

3491

12

444

100

decimal numbers

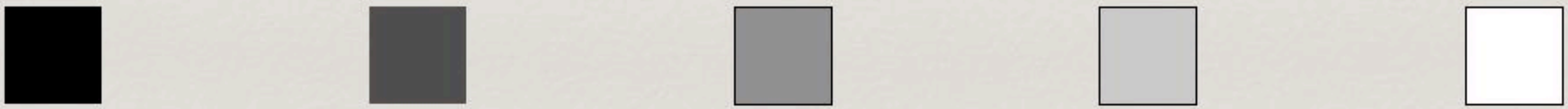
54729

5 4 7 2 9

decimal numbers

54729

$10000 = 10^4$ $1000 = 10^3$ $100 = 10^2$ $10 = 10^1$ $1 = 10^0$

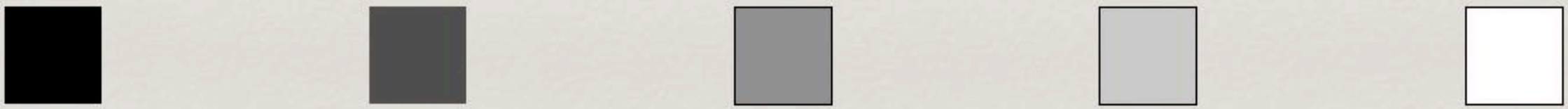


5 4 7 2 9

power notation

54729

$10000 = 10^4$ $1000 = 10^3$ $100 = 10^2$ $10 = 10^1$ $1 = 10^0$



5 4 7 2 9

$= 5 \times 10^4 + 4 \times 10^3 + 7 \times 10^2 + 2 \times 10^1 + 9 \times 10^0$

decimal numbers

0 1 2 3 4 5 6 7 8 9

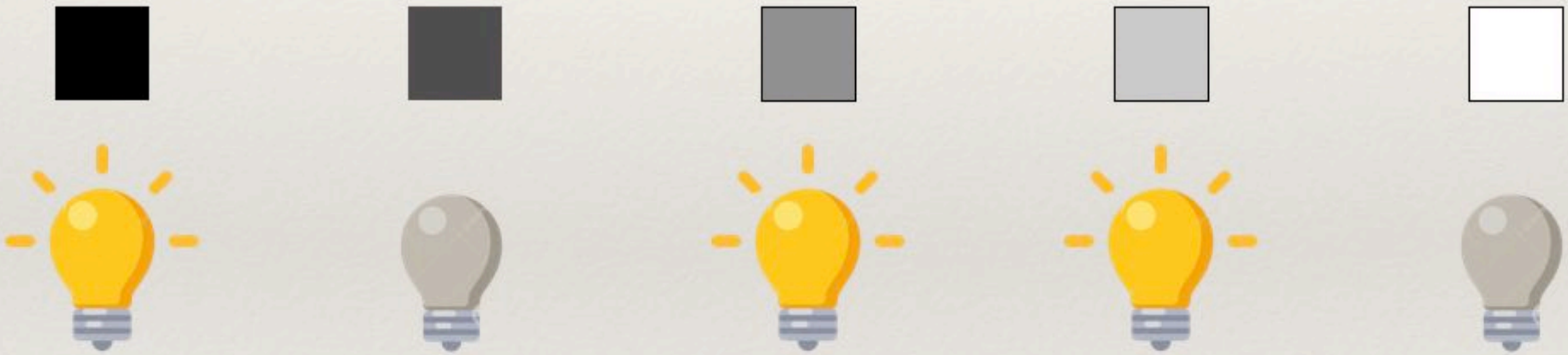
ten symbols

alternative?

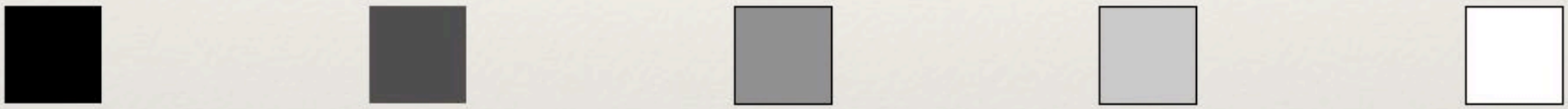
0 1 2 3 4 5 6 7 8 9

0 1

binary numbers



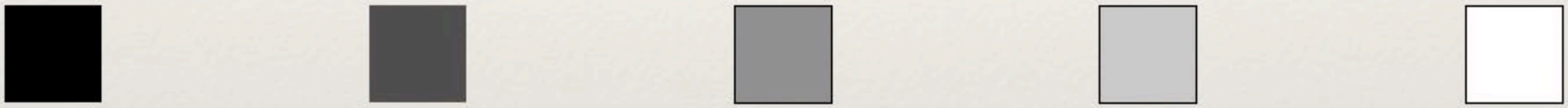
binary numbers



1 0 1 1 0

binary numbers

$2^4 = 16$ $2^3 = 8$ $2^2 = 4$ $2^1 = 2$ $2^0 = 1$



1 0 1 1 0

$= 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0$

$= 16 + 0 + 4 + 2 + 0$

$= 22$

binary numbers

$2^3 = 8$	$2^2 = 4$	$2^1 = 2$	$2^0 = 1$
#	#	#	#

what is the highest value with 4 digits?

binary numbers

$2^3 = 8$	$2^2 = 4$	$2^1 = 2$	$2^0 = 1$
1	1	1	1

what is the highest value with 4 digits?

binary numbers

$2^4 = 16$		$2^3 = 8$		$2^2 = 4$		$2^1 = 2$		$2^0 = 1$
0		1		1		1		1
$= 0 \times 2^4$	+	1×2^3	+	1×2^2	+	1×2^1	+	1×2^0
$= 0$	+	8	+	4	+	2	+	1

$= 15$

binary numbers

$2^4 = 16$		$2^3 = 8$		$2^2 = 4$		$2^1 = 2$		$2^0 = 1$
0		1		1		1		1
$= 0 \times 2^4$	+	1×2^3	+	1×2^2	+	1×2^1	+	1×2^0
$= 0$	+	8	+	4	+	2	+	1

$$= 15$$

$2^4 = 16$	$2^3 = 8$	$2^2 = 4$	$2^1 = 2$	$2^0 = 1$
1	0	0	0	0

bits and bytes

bit = binary digit

1

byte = 8 bites

2^7 2^6 2^5 2^4 2^3 2^2 2^1 2^0

10100110

bits and bytes

little endian

2^7 2^6 2^5 2^4 2^3 2^2 2^1 2^0

1 0 1 0 0 1 1 0

big endian

2^0 2^1 2^2 2^3 2^4 2^5 2^6 2^7

1 0 1 0 0 1 1 0

floating point - precision

scientific notation

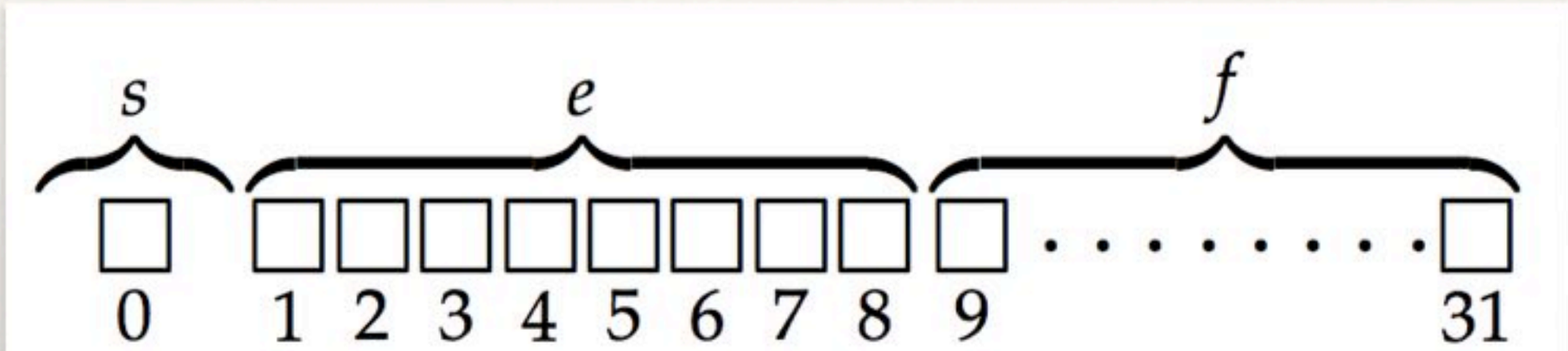
$$100 = 10 \times 10$$

$$100 = 10^2$$

$$500 = 5 \times 100$$

$$500 = 5 \times 10^2$$

floating point



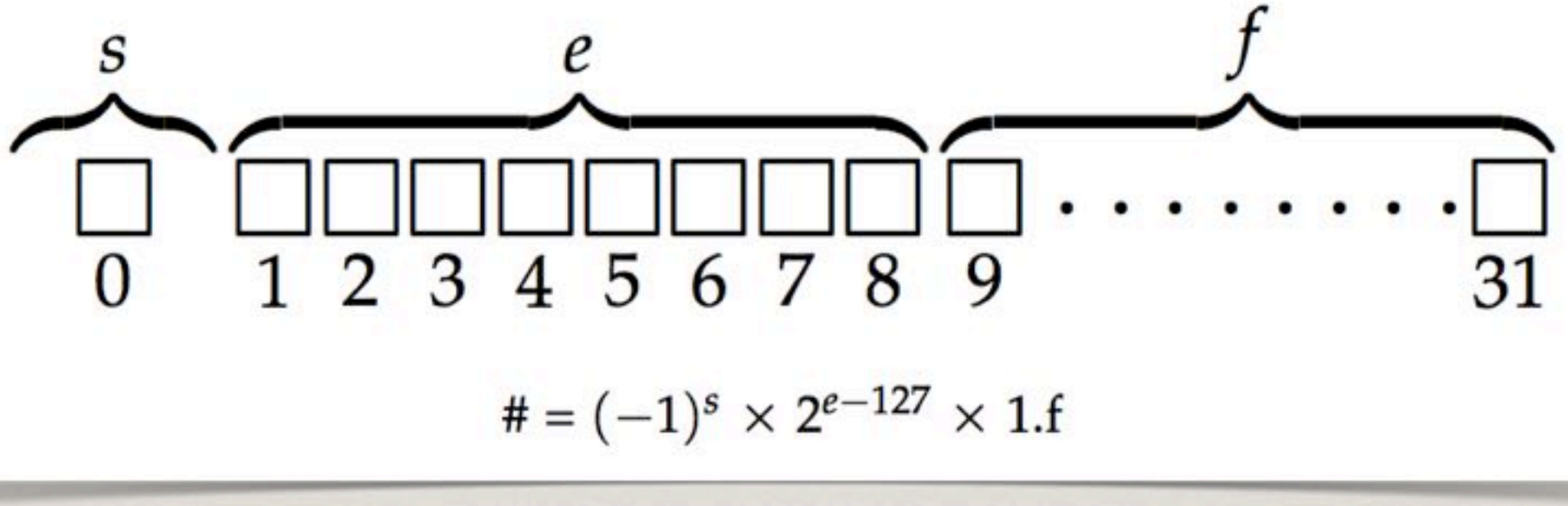
$$\# = (-1)^s \times 2^{e-127} \times 1.\mathrm{f}$$

floating point

- Using the scientific notation, 55.25 is actually represented as $0.5525 * 10^2$
- This leads to some important questions:
  - which is the **biggest** number we can represent?
  - which is the **smallest** number we can represent?
  - what is the **smallest distance** between two consecutive numbers, that we can represent?

Fox p69
Sauer p8

floating point



A floating point number consists of three parts: the **sign** (+ or −), a **mantissa**, which contains the string of significant bits, and an **exponent**. The three parts are stored together in a single computer word.

Sauer p8
Harris p14

floating point

There are three commonly used levels of precision for floating point numbers: single precision, double precision, and extended precision, also known as long-double precision. The number of bits allocated for each floating point number in the three formats is 32,64, and 80, respectively. The bits are divided among the parts as follows:

precision	sign	exponent	mantissa
single	1	8	23
double	1	11	52
long double	1	15	64

Sauer p8

Numerical Analysis

We will study more in details the floating point binaries in a future lecture, when we will study **Numerical Analysis**.

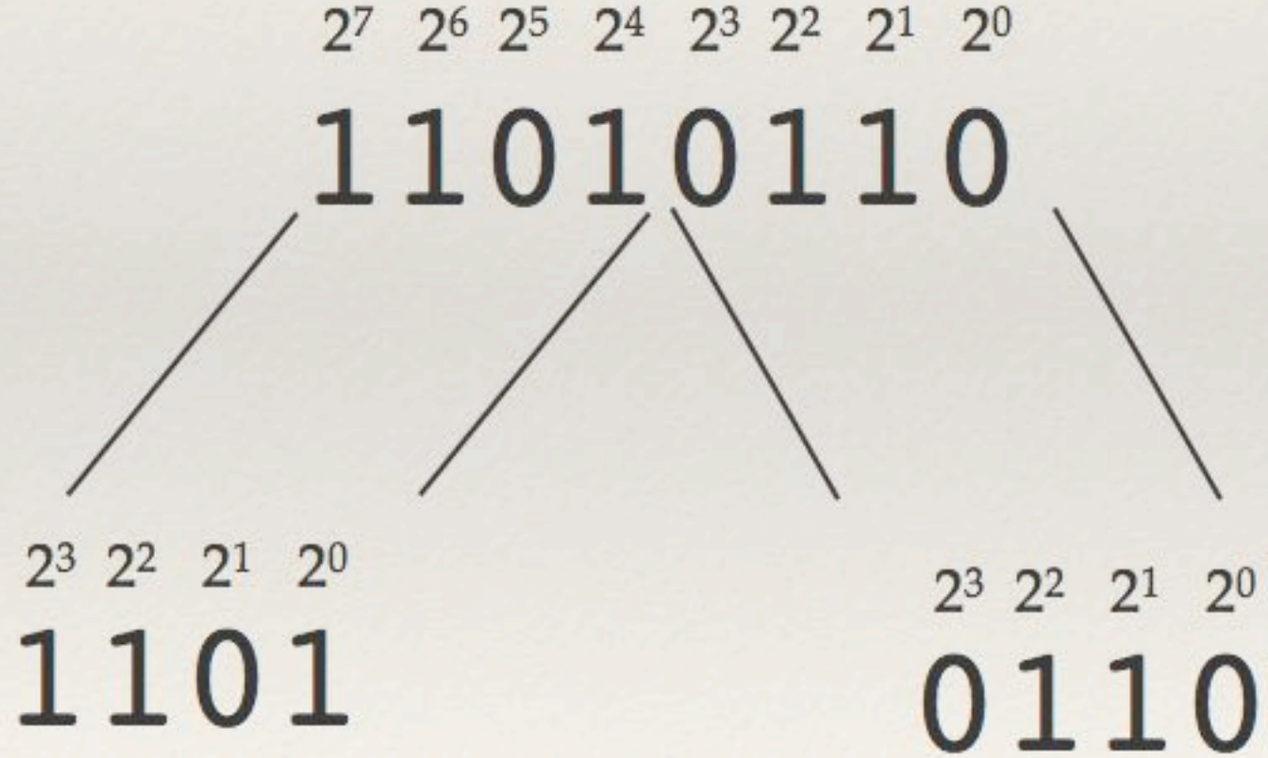
Hexadecimal

bits and bytes

bit = binary digit

1

byte = 8 bites



hexadecimals

half a byte = 4 bites

highest value

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

####

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

$1111_2 = 15_{10}$

hexadecimals

half a byte = 4 bites

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

####

highest value

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

$1111_2 = 15_{10}$

highest value

0123456789ABCDEF => 16^n

hexadecimals

half a byte = 4 bites

$2^3\ 2^2\ 2^1\ 2^0$

####

highest value

$2^3\ 2^2\ 2^1\ 2^0$

$1111_2 = 15_{10}$

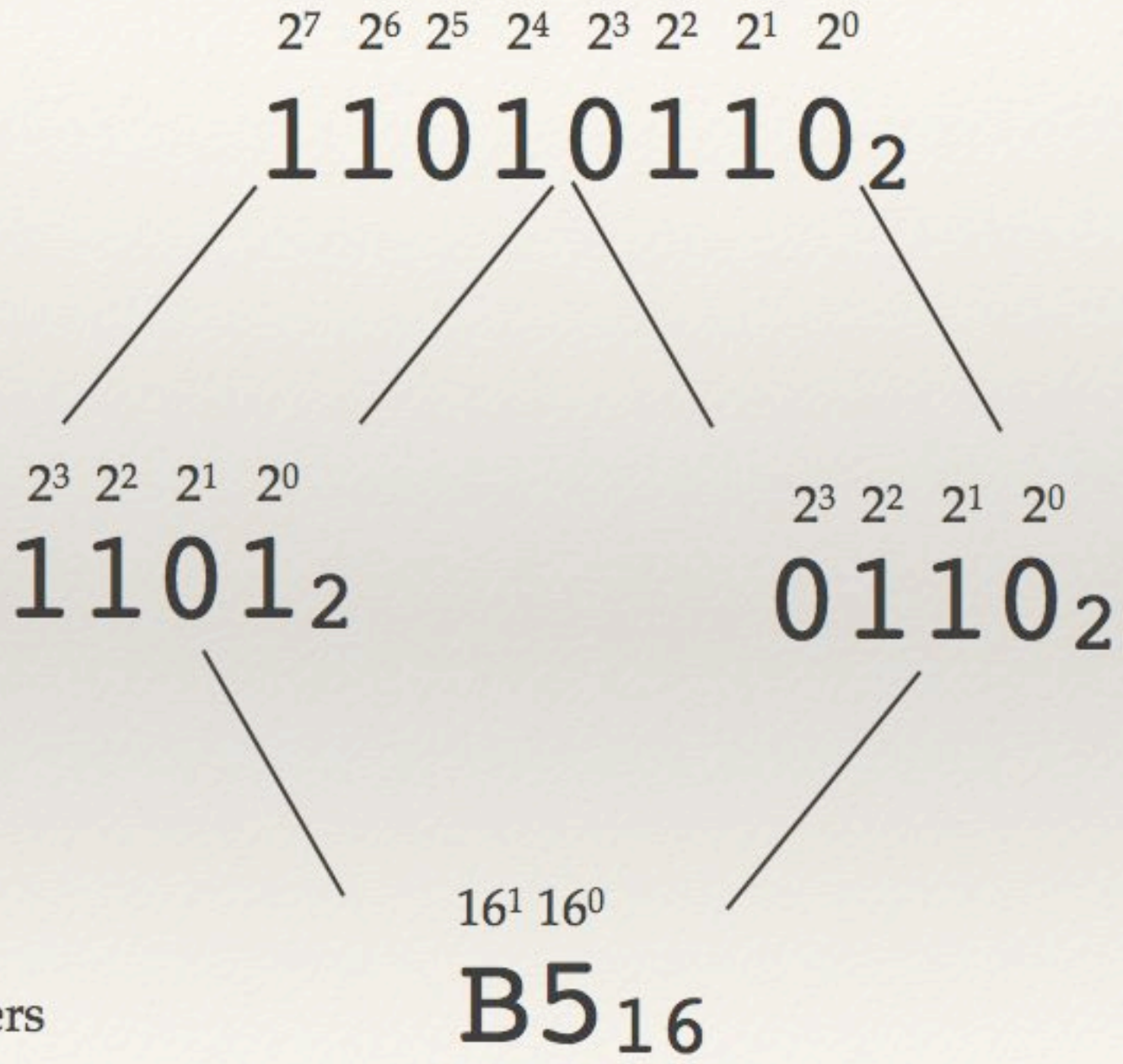
hexadecimal

$16^3\ 16^2\ 16^1\ 16^0$

$####_{16}$

bits and bytes

byte = 8 bites



one byte = two hex numbers

hexadecimals

byte = 8 bites

11010110_2

byte = two hex numbers

$B5_{16}$

hexadecimals - examples

byte = 8 bites

11111111_2

byte = two hex numbers

FF_{16}

hexadecimals - examples

byte = 8 bites

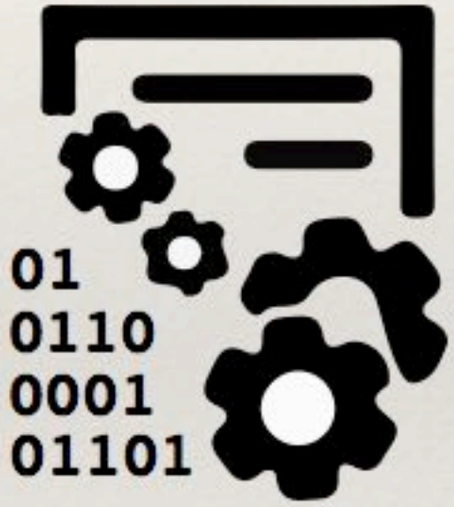
10001000_2

byte = two hex numbers

88_{16}

Files

Binary files



Edit (visualize) Binary files

	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	
0x0000000	50	51	54	54	54	52	00	00	31	2E	30	2E	30	30	00	00	PQTTTR..1.0.00..
0x0000010	46	69	6C	65	5F	47	55	49	44	00	00	00	00	00	00	00	File_GUID.....
0x0000020	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
0x0000030	FF	FF	FF	FF	FF	FF	01	40	28	00	00	00	00	00	00	00	.....@(.....
0x0000040	7B	46	32	41	32	44	30	31	45	2D	33	41	33	39	2D	34	{F2A2D01E-3A39-4
0x0000050	34	41	46	2D	42	30	42	45	2D	32	38	31	39	30	39	33	4AF-B0BE-2819093
0x0000060	33	36	35	42	33	7D	00	00	46	69	6C	65	5F	41	73	73	365B3}..File_Ass
0x0000070	75	72	65	64	43	6F	6E	74	65	6E	74	00	00	00	00	00	uredContent.....
0x0000080	00	00	00	00	00	00	00	00	FF	FF	FF	FF	FF	FF	01	40	.....@
0x0000090	20	00	00	00	00	00	00	00	48	79	64	72	61	48	61	72	.....HydraHar
0x00000A0	70	3A	20	48	57	53	45	54	47	20	53	57	53	45	54	47	p: HWSETG SWSETG
0x00000B0	00	00	00	00	00	00	00	00	43	72	65	61	74	6F	72	53	.....CreatorS
0x00000C0	57	5F	43	6F	6E	74	65	6E	74	56	65	72	73	69	6F	6E	W_ContentVersion
0x00000D0	00	00	00	00	00	00	00	00	FF	FF	FF	FF	FF	FF	01	40	.....@
0x00000E0	08	00	00	00	00	00	00	00	33	2E	30	00	00	00	00	00	.....3.0.....
0x00000F0	43	72	65	61	74	6F	72	53	57	5F	4E	61	6D	65	00	00	CreatorSW_Name..
0x0000100	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
0x0000110	FF	FF	FF	FF	FF	FF	01	40	10	00	00	00	00	00	00	00	.....@.....
0x0000120	48	79	64	72	61	48	61	72	70	20	41	63	71	55	49	00	HydraHarp AcqUI.
0x0000130	43	72	65	61	74	6F	72	53	57	5F	56	65	72	73	69	6F	CreatorSW_Versio
0x0000140	6E	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	n.....
0x0000150	FF	FF	FF	FF	FF	FF	01	40	08	00	00	00	00	00	00	00	.....@.....
0x0000160	33	2E	30	2E	30	2E	31	00	46	69	6C	65	5F	43	72	65	3.0.0.1.File_Cre
0x0000170	61	74	69	6E	67	54	69	6D	65	00	00	00	00	00	00	00	atingTime.....
0x0000180	00	00	00	00	00	00	00	00	FF	FF	FF	FF	08	00	00	21	.....!
0x0000190	9C	A9	D5	45	EC	FE	E4	40	46	69	6C	65	5F	43	6F	6D	.E...@File_Com
0x00001A0	6D	65	6E	74	00	00	00	00	00	00	00	00	00	00	00	00	ment.....
0x00001B0	00	00	00	00	00	00	00	00	FF	FF	FF	FF	FF	FF	01	40	.....@

Start	End	Length	Content
0x00	0x00	0x01	50

Edit (visualize) Binary files

	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	
0x0000000	50	51	54	54	54	52	00	00	31	2E	30	2E	30	30	00	00	PQTTTR..1.0.00..
0x0000010	46	69	6C	65	5F	47	55	49	44	00	00	00	00	00	00	00	File_GUID.......
0x0000020	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....
0x0000030	FF	FF	FF	FF	FF	FF	01	40	28	00	00	00	00	00	00	00	.......@(.......
0x0000040	7B	46	32	41	32	44	30	31	45	2D	33	41	33	39	2D	34	{F2A2D01E-3A39-4
0x0000050	34	41	46	2D	42	30	42	45	2D	32	38	31	39	30	39	33	4AF-B0BE-2819093
0x0000060	33	36	35	42	33	7D	00	00	46	69	6C	65	5F	41	73	73	365B3}..File_Ass
0x0000070	75	72	65	64	43	6F	6E	74	65	6E	74	00	00	00	00	00	uredContent.....
0x0000080	00	00	00	00	00	00	00	00	FF	FF	FF	FF	FF	FF	01	40	...............@
0x0000090	20	00	00	00	00	00	00	00	48	79	64	72	61	48	61	72	.......HydraHar
0x00000A0	70	3A	20	48	57	53	45	54	47	20	53	57	53	45	54	47	p: HWSETG SWSETG
0x00000B0	00	00	00	00	00	00	00	00	43	72	65	61	74	6F	72	53	........CreatorS
0x00000C0	57	5F	43	6F	6E	74	65	6E	74	56	65	72	73	69	6F	6E	W_ContentVersion
0x00000D0	00	00	00	00	00	00	00	00	FF	FF	FF	FF	FF	FF	01	40	...............@
0x00000E0	08	00	00	00	00	00	00	00	33	2E	30	00	00	00	00	00	........3.0.....
0x00000F0	43	72	65	61	74	6F	72	53	57	5F	4E	61	6D	65	00	00	CreatorSW_Name..
0x0000100	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....
0x0000110	FF	FF	FF	FF	FF	FF	01	40	10	00	00	00	00	00	00	00	.......@........
0x0000120	48	79	64	72	61	48	61	72	70	20	41	63	71	55	49	00	HydraHarp AcqUI.
0x0000130	43	72	65	61	74	6F	72	53	57	5F	56	65	72	73	69	6F	CreatorSW_Versio
0x0000140	6E	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	n...............
0x0000150	FF	FF	FF	FF	FF	FF	01	40	08	00	00	00	00	00	00	00	.......@........
0x0000160	33	2E	30	2E	30	2E	31	00	46	69	6C	65	5F	43	72	65	3.0.0.1.File_Cre
0x0000170	61	74	69	6E	67	54	69	6D	65	00	00	00	00	00	00	00	atingTime.......
0x0000180	00	00	00	00	00	00	00	00	FF	FF	FF	FF	08	00	00	21	...............!
0x0000190	9C	A9	D5	45	EC	FE	E4	40	46	69	6C	65	5F	43	6F	6D	.E...@File_Com
0x00001A0	6D	65	6E	74	00	00	00	00	00	00	00	00	00	00	00	00	ment............
0x00001B0	00	00	00	00	00	00	00	00	FF	FF	FF	FF	FF	FF	01	40	...............@

Start	End	Length	Content
0x00	0x05	0x06	50 51 54 54 54 52

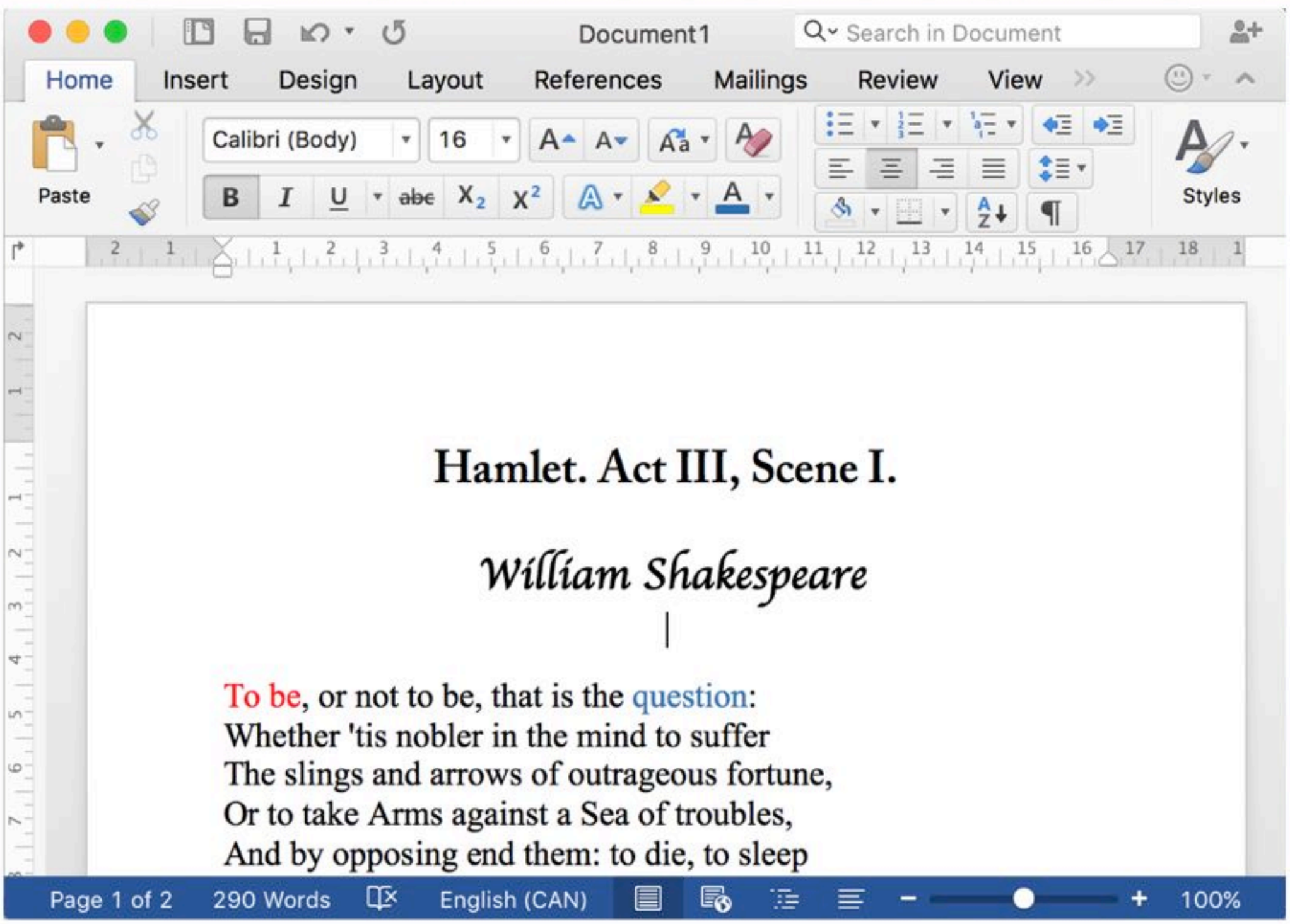
Edit (visualize) Binary files

	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	
0x0000000	50	51	54	54	54	52	00	00	31	2E	30	2E	30	30	00	00	PQTTTR..1.0.00..
0x0000010	46	69	6C	65	5F	47	55	49	44	00	00	00	00	00	00	00	File_GUID.......
0x0000020	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....
0x0000030	FF	FF	FF	FF	FF	FF	01	40	28	00	00	00	00	00	00	00	.......@(.......
0x0000040	7B	46	32	41	32	44	30	31	45	2D	33	41	33	39	2D	34	{F2A2D01E-3A39-4
0x0000050	34	41	46	2D	42	30	42	45	2D	32	38	31	39	30	39	33	4AF-B0BE-2819093
0x0000060	33	36	35	42	33	7D	00	00	46	69	6C	65	5F	41	73	73	365B3}..File_Ass
0x0000070	75	72	65	64	43	6F	6E	74	65	6E	74	00	00	00	00	00	uredContent.....
0x0000080	00	00	00	00	00	00	00	00	FF	FF	FF	FF	FF	FF	01	40	...............@
0x0000090	20	00	00	00	00	00	00	00	48	79	64	72	61	48	61	72	.......HydraHar
0x00000A0	70	3A	20	48	57	53	45	54	47	20	53	57	53	45	54	47	p: HWSETG SWSETG
0x00000B0	00	00	00	00	00	00	00	00	43	72	65	61	74	6F	72	53	........CreatorS
0x00000C0	57	5F	43	6F	6E	74	65	6E	74	56	65	72	73	69	6F	6E	W_ContentVersion
0x00000D0	00	00	00	00	00	00	00	00	FF	FF	FF	FF	FF	FF	01	40	...............@
0x00000E0	08	00	00	00	00	00	00	00	33	2E	30	00	00	00	00	00	........3.0.....
0x00000F0	43	72	65	61	74	6F	72	53	57	5F	4E	61	6D	65	00	00	CreatorSW_Name..
0x0000100	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....
0x0000110	FF	FF	FF	FF	FF	FF	01	40	10	00	00	00	00	00	00	00	.......@........
0x0000120	48	79	64	72	61	48	61	72	70	20	41	63	71	55	49	00	HydraHarp AcqUI.
0x0000130	43	72	65	61	74	6F	72	53	57	5F	56	65	72	73	69	6F	CreatorSW_Versio
0x0000140	6E	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	n...............
0x0000150	FF	FF	FF	FF	FF	FF	01	40	08	00	00	00	00	00	00	00	.......@........
0x0000160	33	2E	30	2E	30	2E	31	00	46	69	6C	65	5F	43	72	65	3.0.0.1.File_Cre
0x0000170	61	74	69	6E	67	54	69	6D	65	00	00	00	00	00	00	00	atingTime.......
0x0000180	00	00	00	00	00	00	00	00	FF	FF	FF	FF	08	00	00	21	...............!
0x0000190	9C	A9	D5	45	EC	FE	E4	40	46	69	6C	65	5F	43	6F	6D	.E...@File_Com
0x00001A0	6D	65	6E	74	00	00	00	00	00	00	00	00	00	00	00	00	ment............
0x00001B0	00	00	00	00	00	00	00	00	FF	FF	FF	FF	FF	FF	01	40	...............@

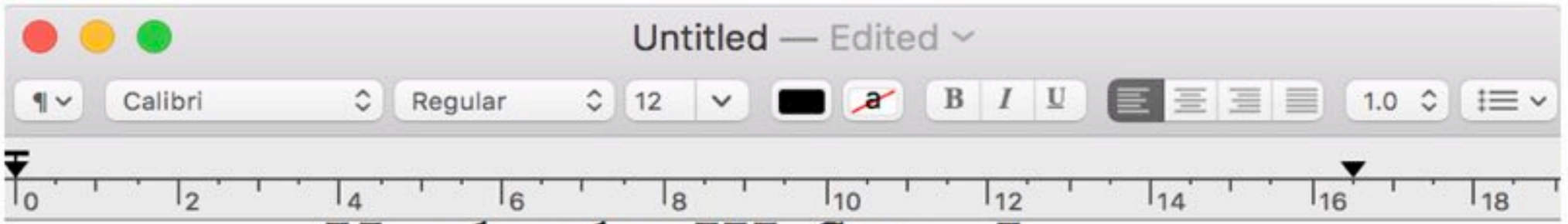
Start	End	Length	Content
0x20	0x26	0x07	00 00 00 00 00 00 00

Text encodings

text files


Document1
Search in Document
Home
Insert
Design
Layout
References
Mailings
Review
View
Paste
Calibri (Body)
16
Styles
Hamlet. Act III, Scene I.
William Shakespeare
To be, or not to be, that is the question:
Whether 'tis nobler in the mind to suffer
The slings and arrows of outrageous fortune,
Or to take Arms against a Sea of troubles,
And by opposing end them: to die, to sleep
Page 1 of 2
290 Words
English (CAN)
100%

text files



Hamlet. Act III, Scene I.

William Shakespeare

To be, or not to be, that is the question:
Whether 'tis nobler in the mind to suffer
The slings and arrows of outrageous fortune,
Or to take Arms against a Sea of troubles,
And by opposing end them: to die, to sleep
No more; and by a sleep, to say we end
the heart-ache, and the thousand natural shocks
that Flesh is heir to? 'Tis a consummation
devoutly to be wished. To die, to sleep,
To sleep, perchance to Dream: aye, there's the rub

text files

Hamlet.txt

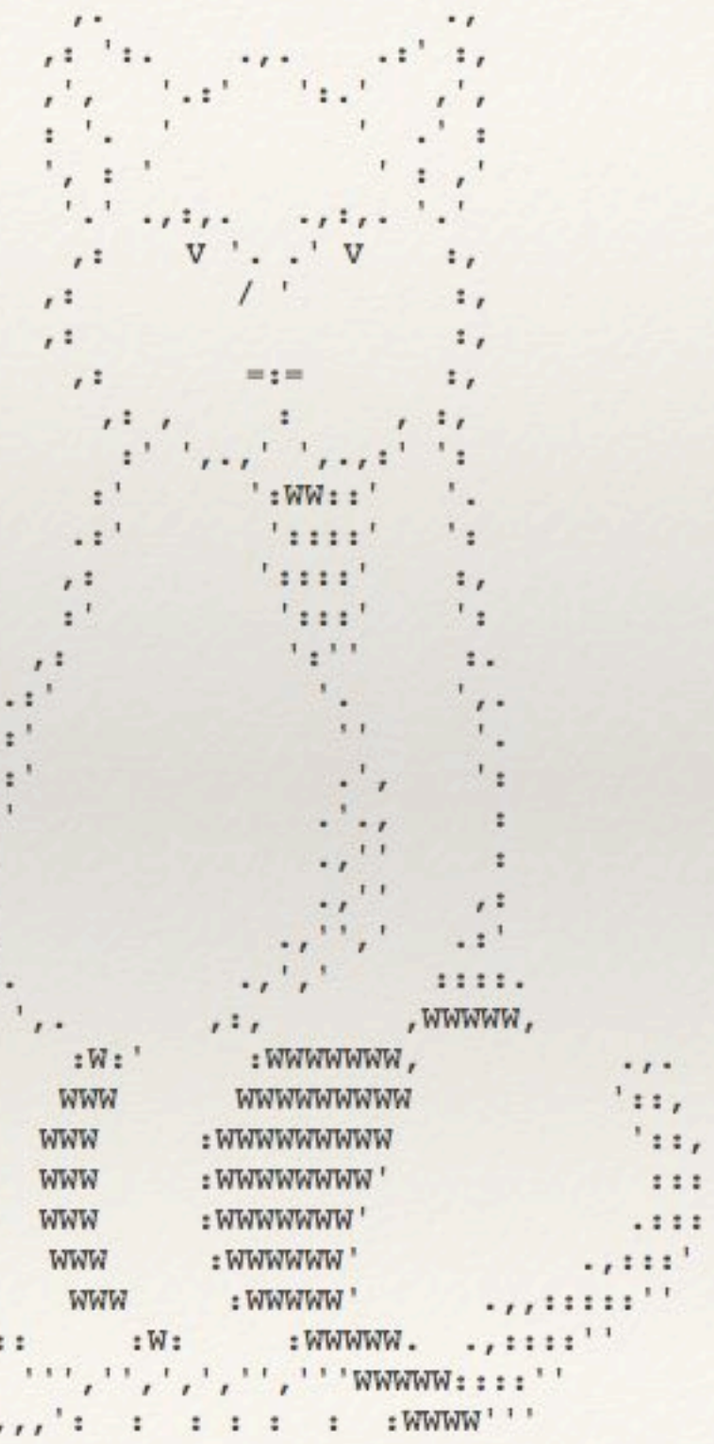
```
Hamlet. Act III, Scene I.

 William Shakespeare


To be, or not to be, that is the question:
Whether 'tis nobler in the mind to suffer
The slings and arrows of outrageous fortune,
Or to take Arms against a Sea of troubles,
And by opposing end them: to die, to sleep
No more; and by a sleep, to say we end
the heart-ache, and the thousand natural shocks
that Flesh is heir to? 'Tis a consummation
devoutly to be wished. To die, to sleep,
To sleep, perchance to Dream; aye, there's the rub,
for in that sleep of death, what dreams may come,
when we have shuffled off this mortal coil,
must give us pause. There's the respect
that makes Calamity of so long life:
For who would bear the Whips and Scorns of time,
the Oppressor's wrong, the proud man's Contumely,
the pangs of despised Love, the Law’s delay,
the insolence of Office, and the spurns
that patient merit of the unworthy takes,
when he himself might his Quietus make
```

ascii characters

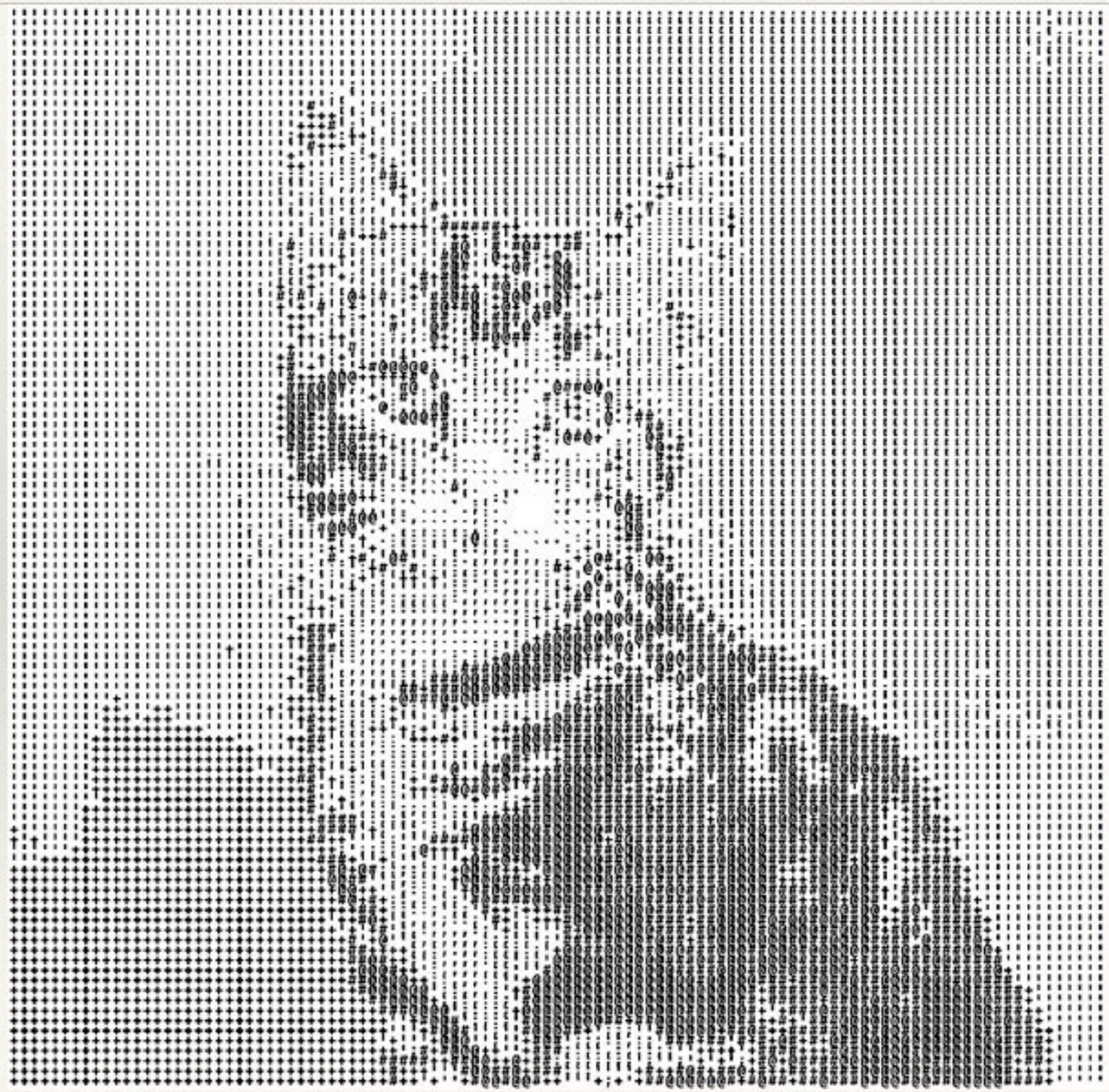
American Standard Code for Information Interchange (ASCII)

010 0000	32	space	100 0001	65	A	110 0001	97	a
010 0001	33	!	100 0010	66	B	110 0010	98	b
010 0010	34	"	100 0011	67	C	110 0011	99	c
010 0011	35	#	100 0100	68	D	110 0100	100	d
010 0100	36	$	100 0101	69	E	110 0101	101	e
010 0101	37	%	100 0110	70	F	110 0110	102	f
010 0110	38	&	100 0111	71	G	110 0111	103	g
010 0111	39	'	100 1000	72	H	110 1000	104	h
010 1000	40	(	100 1001	73	I	110 1001	105	i
010 1001	41	)	100 1010	74	J	110 1010	106	j
010 1010	42	*	100 1011	75	K	110 1011	107	k
010 1011	43	+	100 1100	76	L	110 1100	108	l
010 1100	44	,	100 1101	77	M	110 1101	109	m
010 1101	45	-	100 1110	78	N	110 1110	110	n
010 1110	46	.	100 1111	79	O	110 1111	111	o
010 1111	47	/	101 0000	80	P	111 0000	112	p
011 0000	48	0	101 0001	81	Q	111 0001	113	q
011 0001	49	1	101 0010	82	R	111 0010	114	r
011 0010	50	2	101 0011	83	S	111 0011	115	s
011 0011	51	3	101 0100	84	T	111 0100	116	t
011 0100	52	4	101 0101	85	U	111 0101	117	u
011 0101	53	5	101 0110	86	V	111 0110	118	v
011 0110	54	6	101 0111	87	W	111 0111	119	w
011 0111	55	7	101 1000	88	X	111 1000	120	x
011 1000	56	8	101 1001	89	Y	111 1001	121	y
011 1001	57	9	101 1010	90	Z	111 1010	122	z

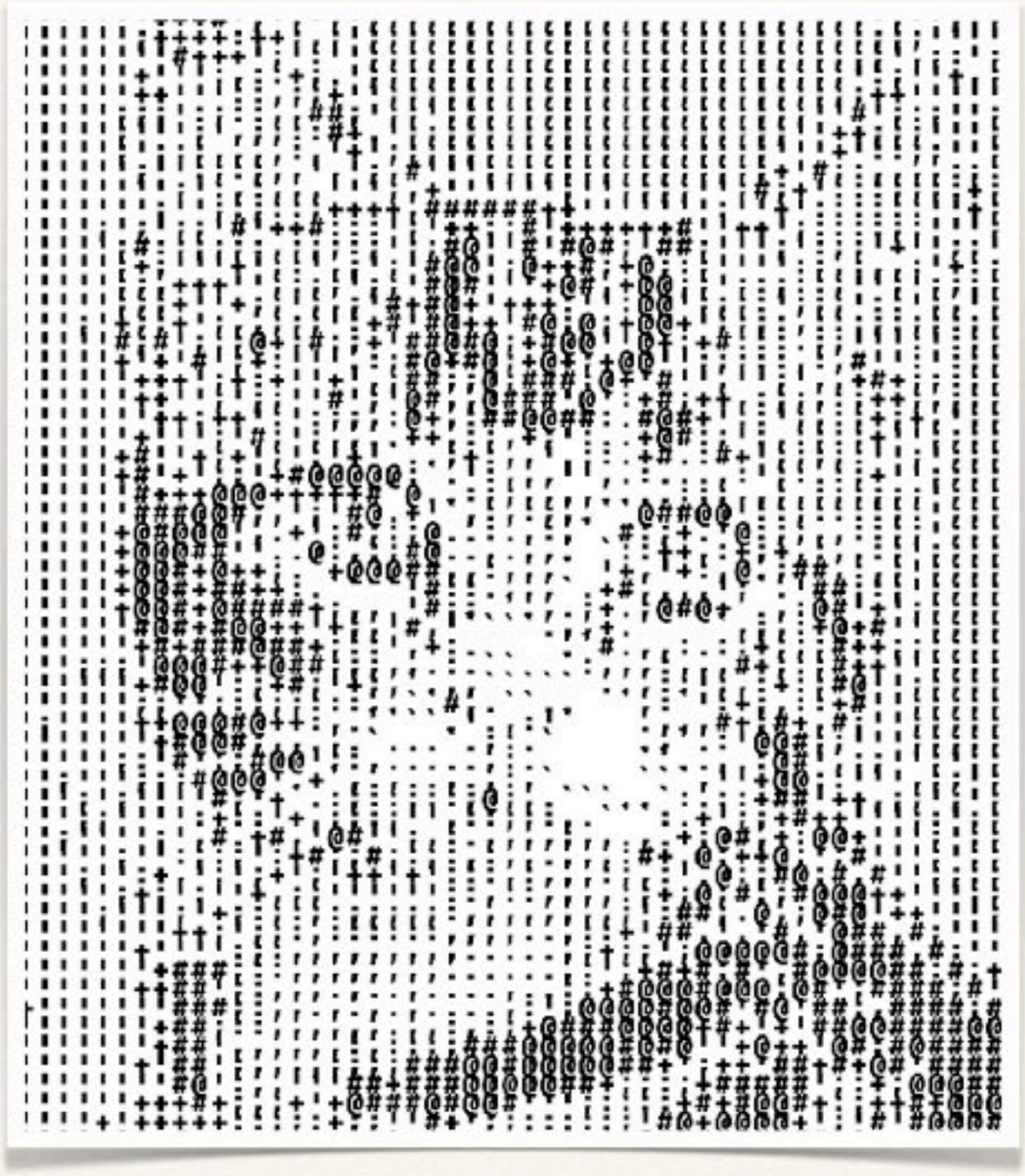
ascii characters



ascii characters



ascii characters



ascii encoding

- The best known and most widely used character encoding standard is the **American Standard Code for Information Interchange** (ASCII).
- The first version of ASCII was published in **1964** as a standard way of representing textual data in computer memory and sending it over communication links between computers.
- ASCII is based on a **seven-bit byte**. Each byte represented a character, and characters were represented by assigning them to individual binary numbers.

Gillam p7

ascii encoding

what is the highest value that we can write with 7 binary digits?

$2^6\ 2^5\ 2^4\ 2^3\ 2^2\ 2^1\ 2^0$

1111111

$2^7\ 2^6\ 2^5\ 2^4\ 2^3\ 2^2\ 2^1\ 2^0$

10000000

$$2^7 = 128$$

Gillam p7

ascii encoding

- Perhaps the main deficiency in ASCII comes from the "A" in its name: **American**. ASCII is an American standard, and was designed for the storage and transmission of English text. 95 characters are sufficient for representing English text, barely, but that's it. On early teletype machines, ASCII could also be used to represent the accented letters found in many European languages, but this capability disappeared in the transition from teletypes to CRT terminals.

CRT = Cathode Ray Tube

Gillam p7

ascii encoding

- Perhaps the main deficiency in ASCII comes from the "A" in its name: **American**. ASCII is an American standard, and was designed for the storage and transmission of English text. 95 characters are sufficient for representing English text, barely, but that's it. On early teletype machines, ASCII could also be used to represent the accented letters found in many European languages, but this capability disappeared in the transition from teletypes to CRT terminals.
- So, as computer use became more and more widespread in different parts of the world, alternative methods of representing characters in computers arose for representing other languages, leading to the situation we have today, where there are generally three or four different encoding schemes for every language and writing system in use today.

Gillam p7

Unicode

- Unicode is the latest of several attempts to solve this Tower of Babel problem by creating a universal character encoding.
- Its main way of doing this is to increase the size of the possible encoding space by **increasing the number of bits used to encode each character**.
- Most other character encodings are based upon an **eight-bit byte**, which provides enough space to encode a maximum of **256 characters** (in practice, most encodings reserve some of these values for control signals and encode fewer than 256 characters).

Gillam p7

Unicode

- Unicode uses a 16-bit word to encode characters, allowing up to 65,536 characters to be encoded. 65,000 characters, with careful management, is enough to allow encoding of the vast majority of characters in the vast majority of written languages in use today.
- The current version of Unicode, version 3.2, actually encodes 95,156 different characters—it actually does use a scheme to represent the less-common characters using two 16-bit units, but with 50,212 characters actually encoded using only a single unit, you rarely encounter the two-unit characters. In fact, these 50,212 characters include all of the characters representable with all of the other character encoding methods that are in reasonably widespread use.

Gillam p7

UTF-8

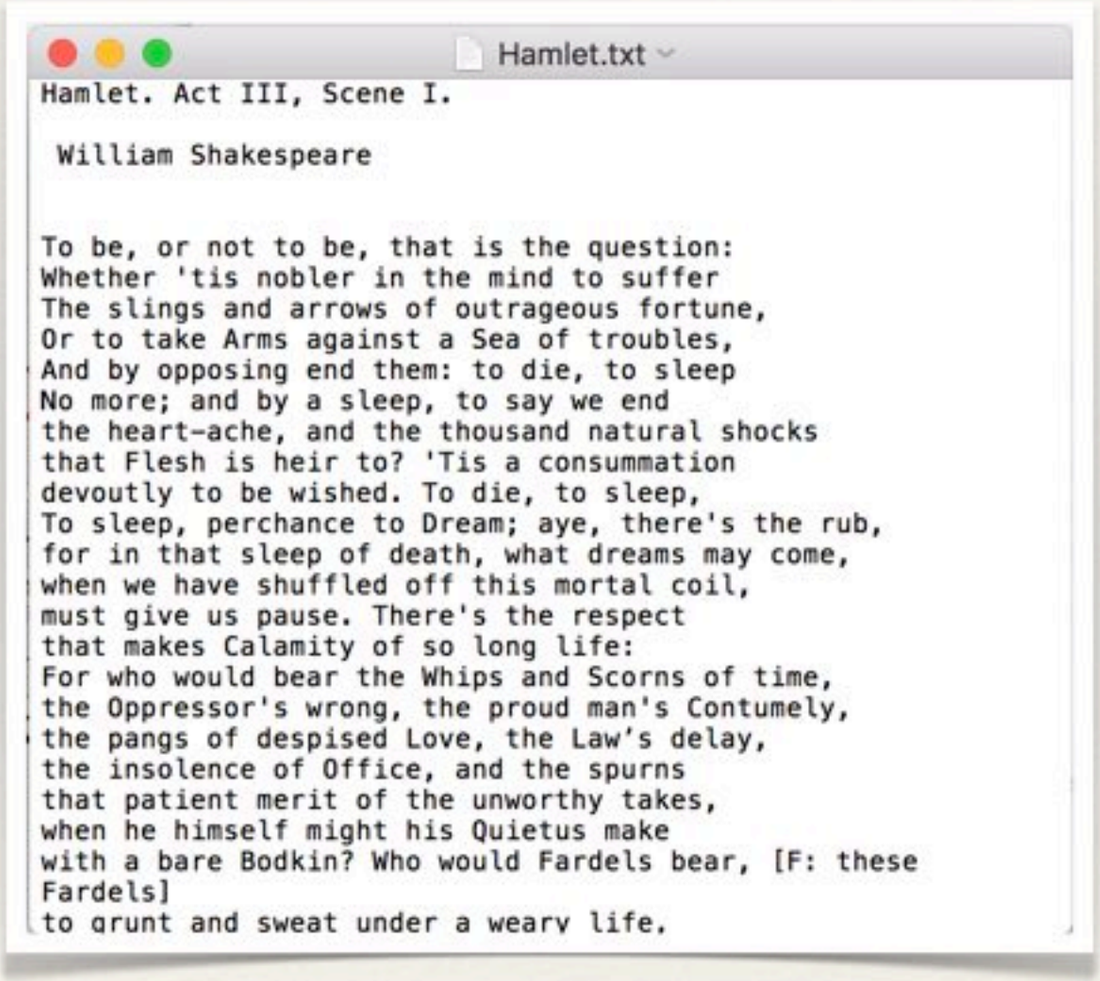
- UTF-8 is a variable width character encoding capable of encoding all 1,112,064 valid code points in **Unicode** using one to four 8-bit **bytes**.
- The name is derived from Unicode Transformation Format – 8-bit.

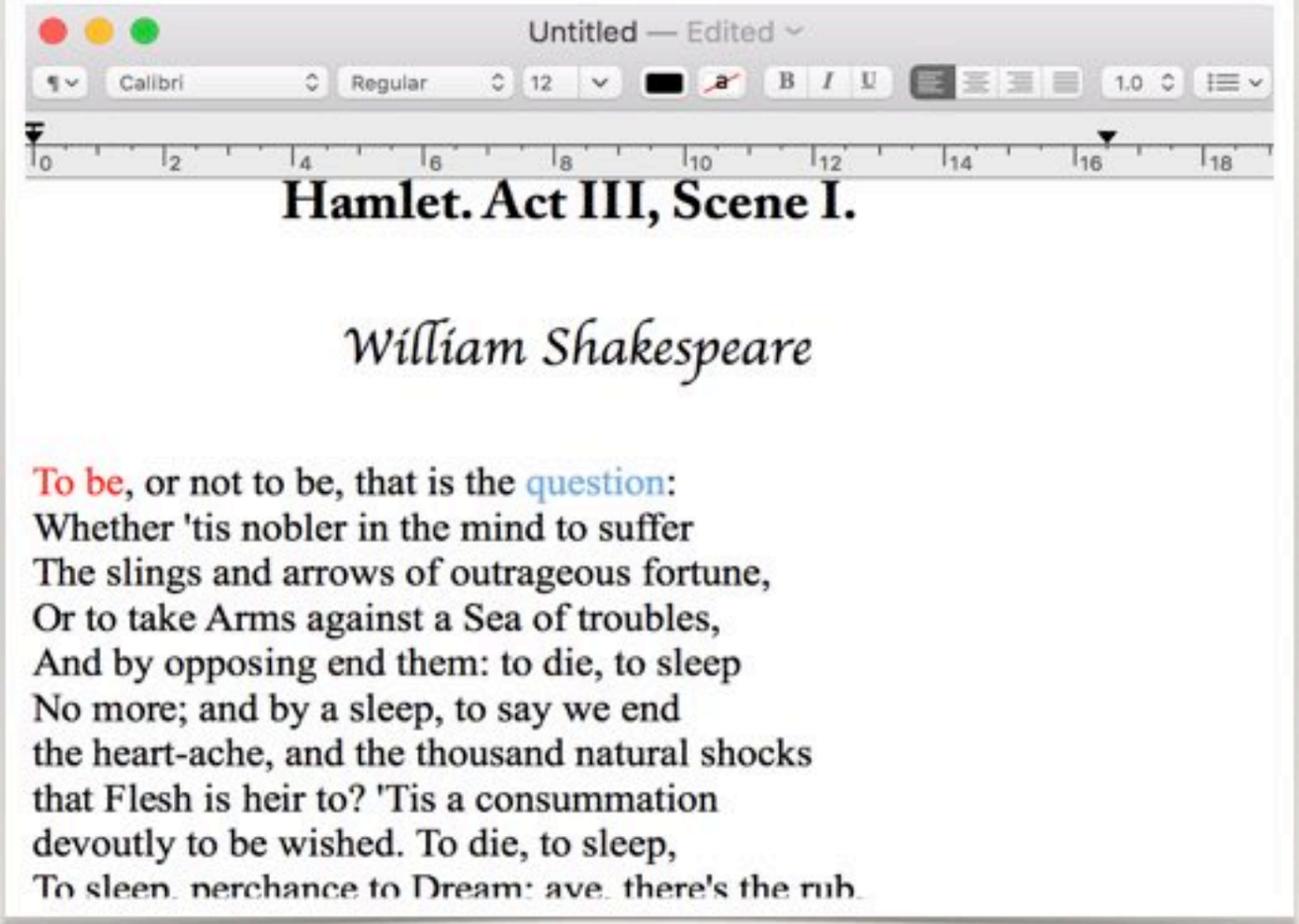
Wikipedia

cyrillic

0410 А	0430	а
0411 Б	0431	б
0412 В	0432	в
0413 Г	0433	г
0414 Д	0434	д
0415 Е	0435	е
0416 Ж	0436	ж
0417 З	0437	з
0418 И	0438	и
0419 Й	0439	й
041AК	043A	к
041B Л	043B	л
041CМ	043C	м
041DН	043D	н
041E О	043E	о
041F П	043F	п
0420 Р	0440	р
0421 С	0441	с
0422 Т	0442	т
0423 У	0443	у
0424 Ф	0444	ф
0425 Х	0445	х
0426 Ц	0446	ц
0427 Ч	0447	ч
0428 Ш	0448	ш
0429 Щ	0449	щ
042AЪ	044A	ъ
042B Ы	044B	ы
042CЬ	044C	ь
042DЭ	044D	э
042E Ю	044E	ю
042F Я	044F	я

.txt vs .rtf


Hamlet.txt
Hamlet. Act III, Scene I.
William Shakespeare
To be, or not to be, that is the question:
Whether 'tis nobler in the mind to suffer
The slings and arrows of outrageous fortune,
Or to take Arms against a Sea of troubles,
And by opposing end them: to die, to sleep
No more; and by a sleep, to say we end
the heart-ache, and the thousand natural shocks
that Flesh is heir to? 'Tis a consummation
devoutly to be wished. To die, to sleep,
To sleep, perchance to Dream; aye, there's the rub,
for in that sleep of death, what dreams may come,
when we have shuffled off this mortal coil,
must give us pause. There's the respect
that makes Calamity of so long life:
For who would bear the Whips and Scorns of time,
the Oppressor's wrong, the proud man's Contumely,
the pangs of despised Love, the Law's delay,
the insolence of Office, and the spurns
that patient merit of the unworthy takes,
when he himself might his Quietus make
with a bare Bodkin? Who would Fardels bear, [F: these
Fardels]
to grunt and sweat under a weary life,


Untitled — Edited
Calibri
Regular
12
1.0
Hamlet. Act III, Scene I.
William Shakespeare
To be, or not to be, that is the question:
Whether 'tis nobler in the mind to suffer
The slings and arrows of outrageous fortune,
Or to take Arms against a Sea of troubles,
And by opposing end them: to die, to sleep
No more; and by a sleep, to say we end
the heart-ache, and the thousand natural shocks
that Flesh is heir to? 'Tis a consummation
devoutly to be wished. To die, to sleep,
To sleep, perchance to Dream; aye, there's the rub,

Rich Text Format

- It's also important to keep in mind what Unicode isn't. First, Unicode is a standard scheme for representing plain text in computers and data communication. It is not a scheme for representing rich text (sometimes called "fancy text" or "styled text"). This is an important distinction. Plain text is the words, sentences, numbers, and so forth themselves. Rich text is plain text plus information about the text, especially information on the text's visual presentation (e.g., the fact that a given word is in italics), the structure of a document (e.g., the fact that a piece of text is a section header or footnote), or the language (e.g., the fact that a particular sentence is in Spanish). Rich text may also include non-text items that travel with the text, such as pictures.
- It can be somewhat tough to draw a line between what qualifies as plain text, and therefore should be encoded in Unicode, and what's really rich text. In fact, debates on this very subject flare up from time to time in the various Unicode discussion forums. The basic rule is that plain text contains all of the information necessary to carry the semantic meaning of the text—the letters, spaces, digits, punctuation, and so forth. If removing it would make the text unintelligible, then it's plain text.

Gillam p7