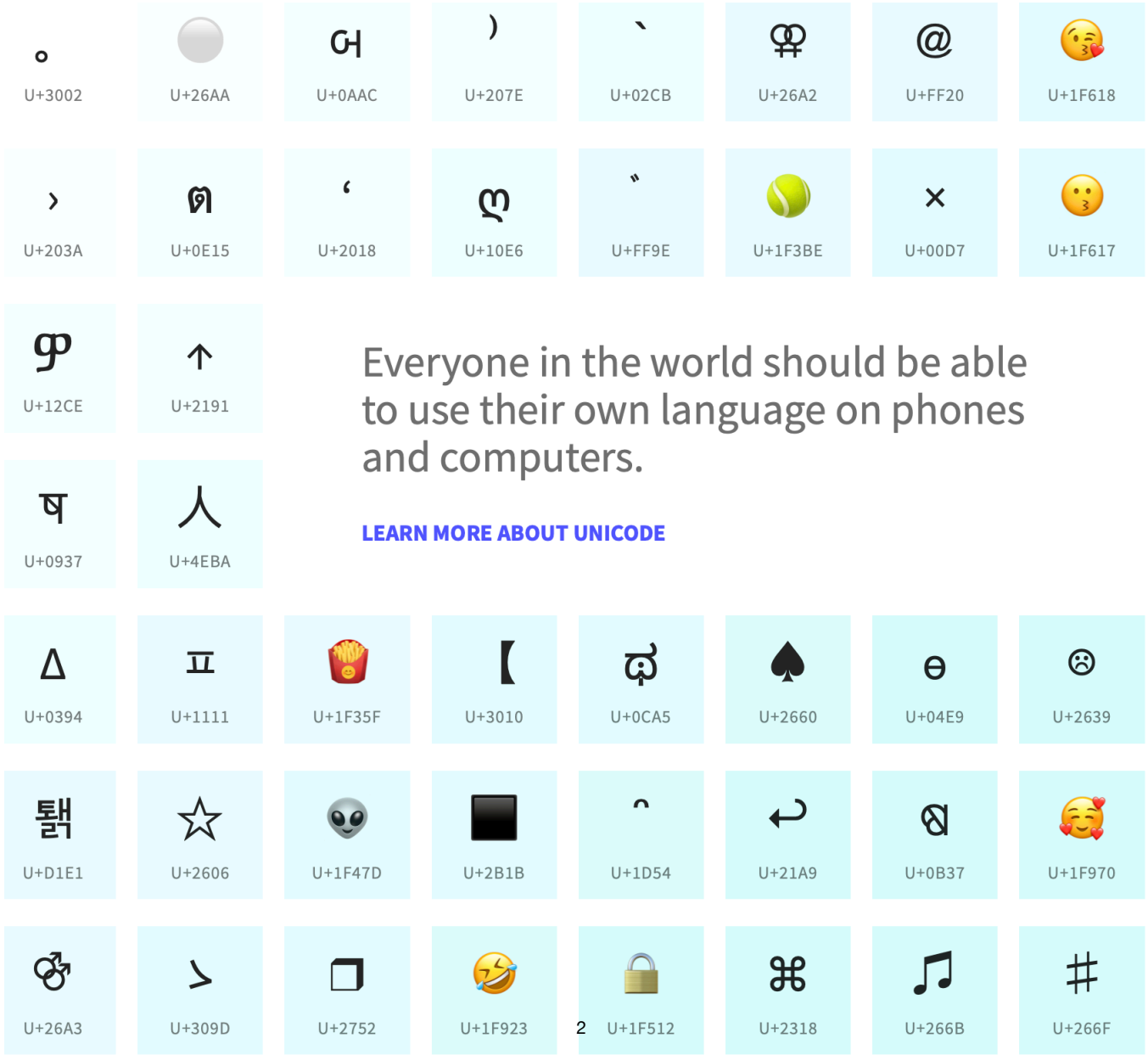
Introduction To Unicode

Making the digital world more inclusive

Alexandre Bergel
Computer Science Department - FCFM
University of Chile
http://bergel.eu **abergel@dcc.uchile.cl** **@AlexBergel**


U+3002
U+26AA
U+0AAC
U+207E
U+02CB
U+26A2
U+FF20
U+1F618
U+203A
U+0E15
U+2018
U+10E6
U+FF9E
U+1F3BE
U+00D7
U+1F617
U+12CE
U+2191
Everyone in the world should be able to use their own language on phones and computers.
LEARN MORE ABOUT UNICODE
U+0937
U+4EBA
U+0394
U+1111
U+1F35F
U+3010
U+0CA5
U+2660
U+04E9
U+2639
U+D1E1
U+2606
U+1F47D
U+2B1B
U+1D54
U+21A9
U+0B37
U+1F970
U+26A3
U+309D
U+2752
U+1F923
U+1F512
U+2318
U+266B
U+266F

https://en.wikipedia.org/wiki/Teleprinter#/media/File:WACsOperateTeletype.jpg

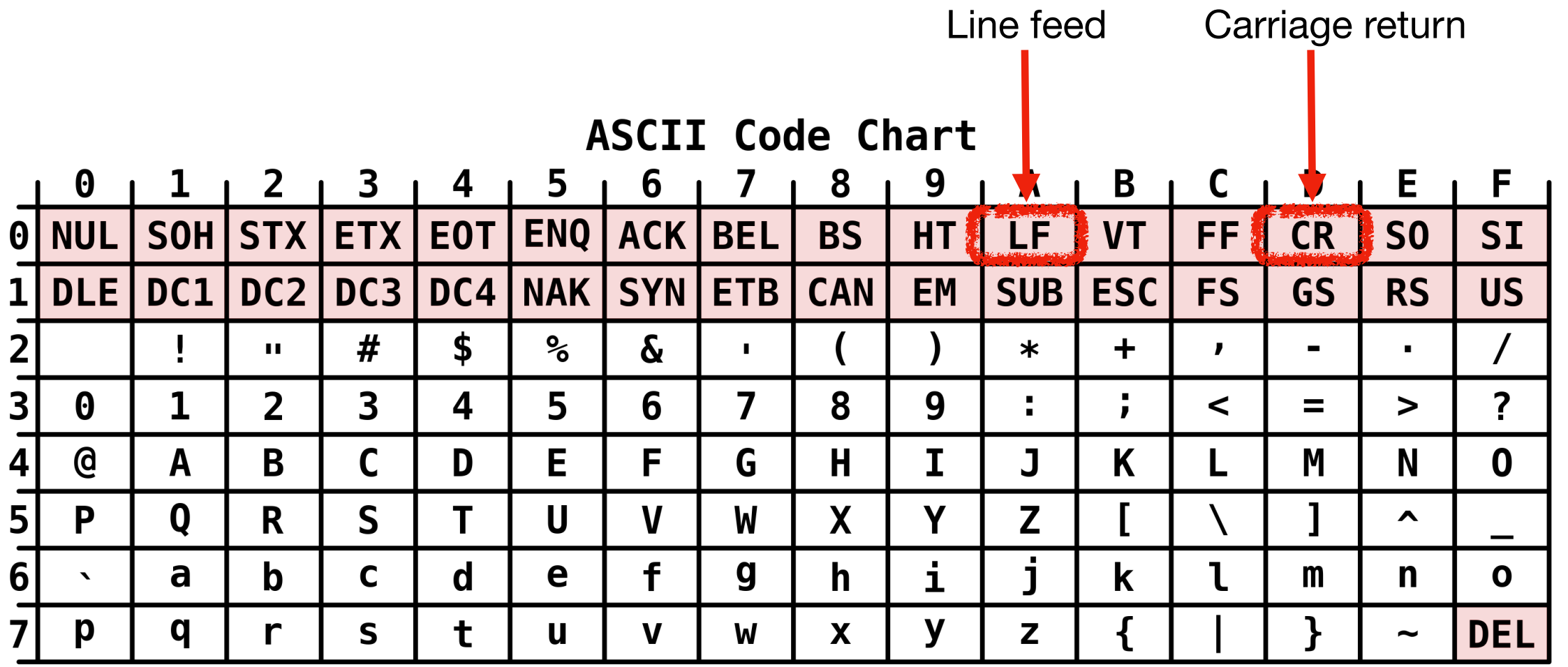
https://en.wikipedia.org/wiki/Teleprinter#/media/File:Siemens_t37h_without_cover.jpg

ASCII

- Supports meaningful exchange of text data
- Proposed in 1963
- Coded on 7-bits => 128 characters
  - A = 65
- Very limited, not even adequate for English
  - e.g., “résumé” is an English word
- Only letters, digits, and punctuation are considered as printable characters

ASCII Code Chart

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
1	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2		!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	DEL



	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
1	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2		!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	DEL

“Hello\nWorld” → Hello
World

ASCII Code Chart

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
1	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2		!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	DEL

“Hello\nWorld”

Hello
World

Line feed

Carriage return

ASCII Code Chart

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
1	DLE	DC1	DC2	DC3									FS	GS	RS	US
2		!	"	#									,	-	.	/
3	0	1	2	3									<	=	>	?
4	@	A	B	C									L	M	N	O
5	P	Q	R	S									\	]	^	_
6	`	a	b	c									l	m	n	o
7	p	q	r	s									|	}	~	DEL

https://www.pxfuel.com/en/free-photo-xcoyx

Many other standards

- ASCII has many limitations
- Many industrials proposed *their own improvement*
  - MacRoman from Apple
  - IBM's EBCDIC-based code pages
  - Microsoft, SAP, Oracle, …

Unicode

- *“Unicode is an information technology standard for the consistent encoding, representation, and handling of text expressed in most of the world's writing systems.”*
- Designed to improve the mess inherited from *telegraph machine*
- *Enable world-wide interchange of data*
- Multilingual
- A single implementation
- Support legacy data

Writing direction

① آ ظ ١١٢٣ صفحة إلى انظر رجاءً.

② נא ראה עמוד 1123.

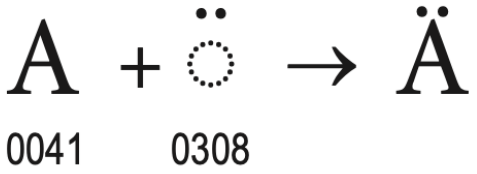
③ راجع صفحة ١١٢٣ من فضلك.

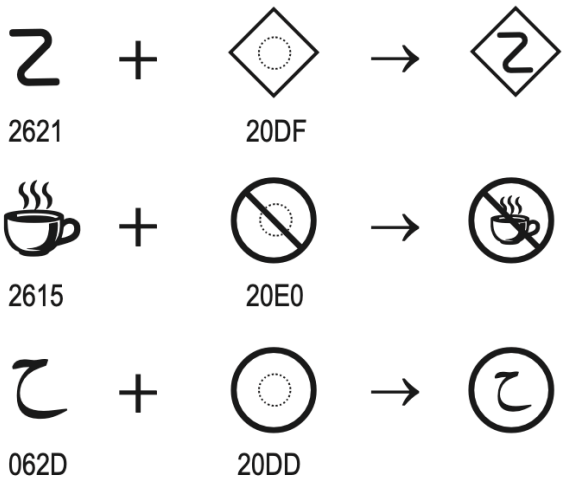
④ Please see page 1123.

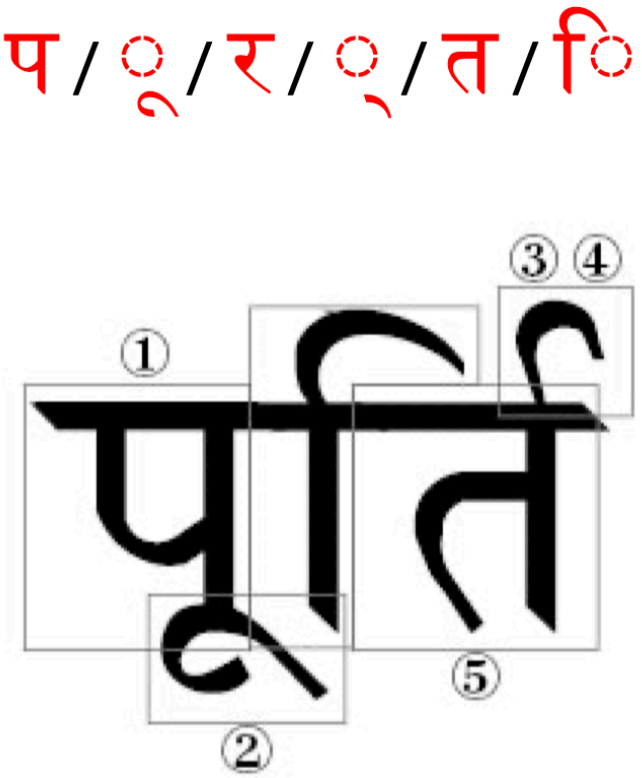
⑤ 1123ページをみてください。

⑥ 1123ページをみてください。

Character composition







Overview of Unicode

Unicode

- > 143,859 characters
- > 154 modern and historical scripts
- *Script:* collection of letters and other written signs used to represent textual information

Character model

- Four layers
- Level 1: Abstract character set => *What is a character?*
- Level 2: Coded character set => *How to name and enumerate abstract character?*
- Level 3: Character encoding forms => *How to represent coded characters in a computer?*
- Level 4: Character encoding schemes => *how to serialize characters into bytes?*

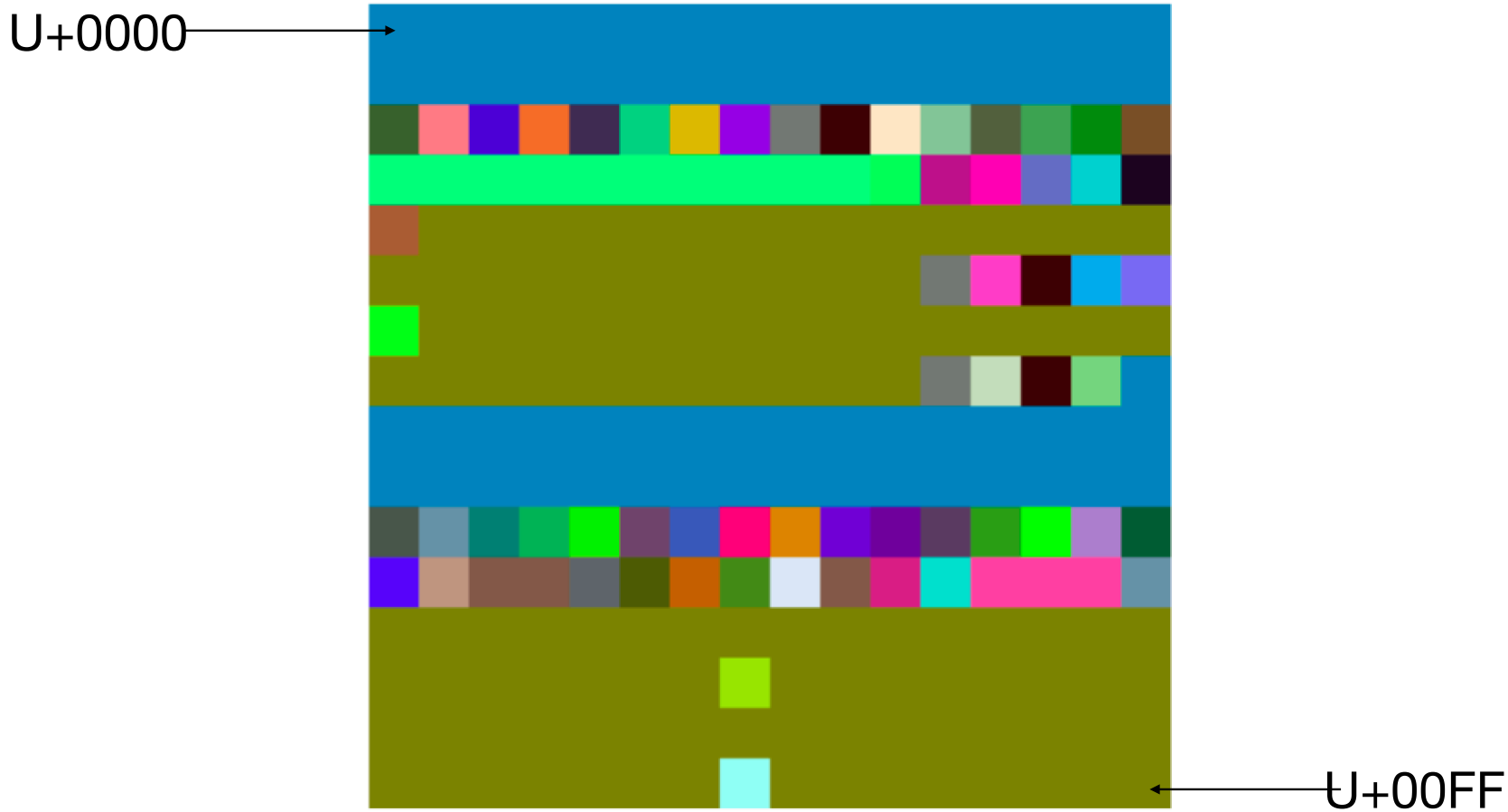
Abstract character set (level 1/4)

- Character: *The smallest component of written language that has semantic value*
- Wide variation across scripts:
  - *Alphabetic* => each character is a letter. Both *consonant* and *vowel* have equal status
  - *Syllabary* => each character is a syllab
    - E.g., Hiragana (あ, せ, ぬ), Cheerokee (Ꮣ, Ꭷ, Ꮷ), Vai (ꕢ, ꖜ, ꖧ)
  - *Abjad* => each character is a consonant, vowel marking is absent
    - E.g., Hebrew (ג, ה, ט), Punic ( , ,)
  - *Abugidas* => each character is a sequence of consonant - vowel, the vowel notation is secondary
  - *Logographic* => each character is a word
    - E.g., Egyptian hieroglyphs, emoji (although still debated)
- Abstract character: *a unit of information used for the organization, control, or representation of textual data*

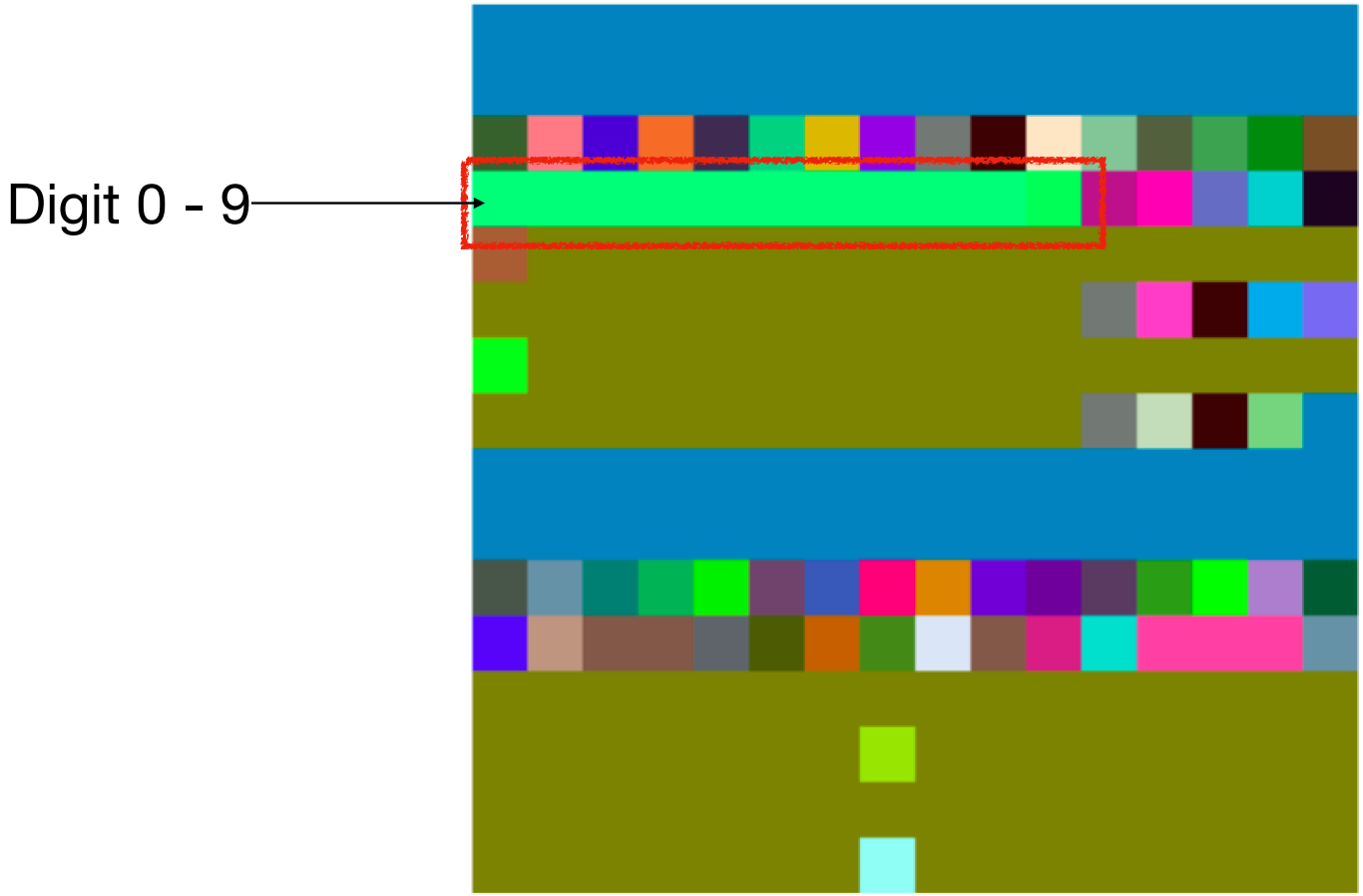
Coded character set (level 2/4)

- Give a name and a code point to each abstract character
- Name: LATIN CAPITAL LETTER A
- Code point: pure number
  - Legal value: U+0000 - U+10FFFF
  - Space for >1M different characters
- Characters that are specific to a script *are mostly grouped*
- *No connection* to the computer

The first 256 characters

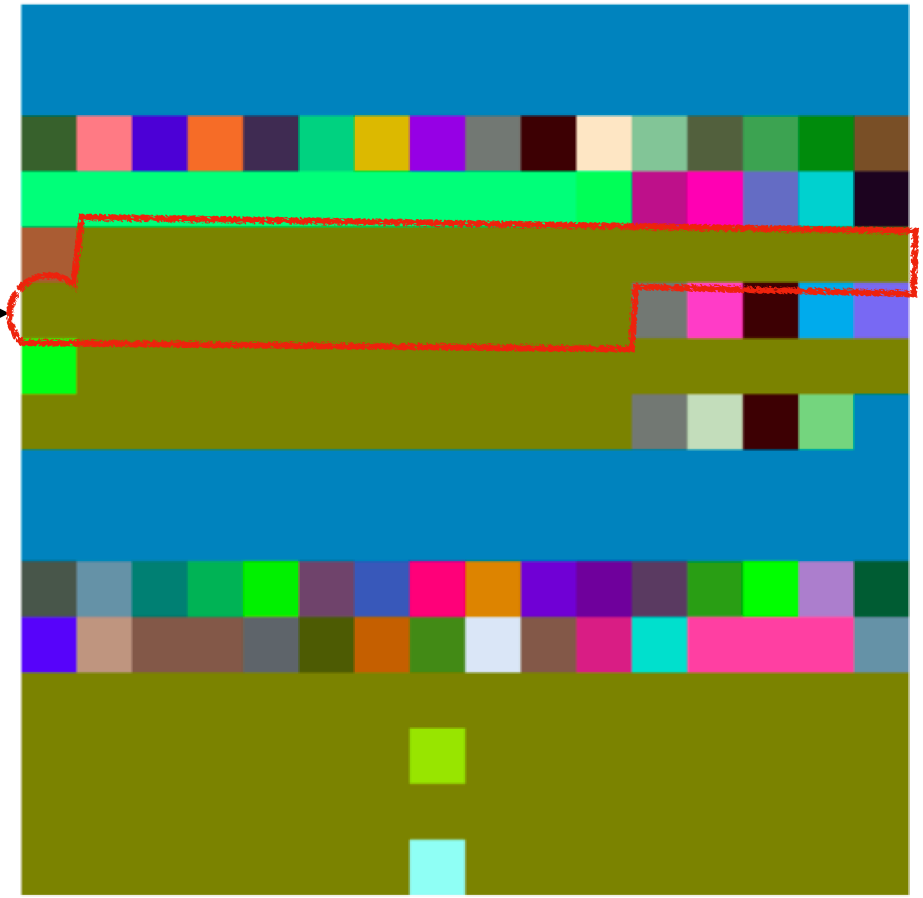


The first 256 characters



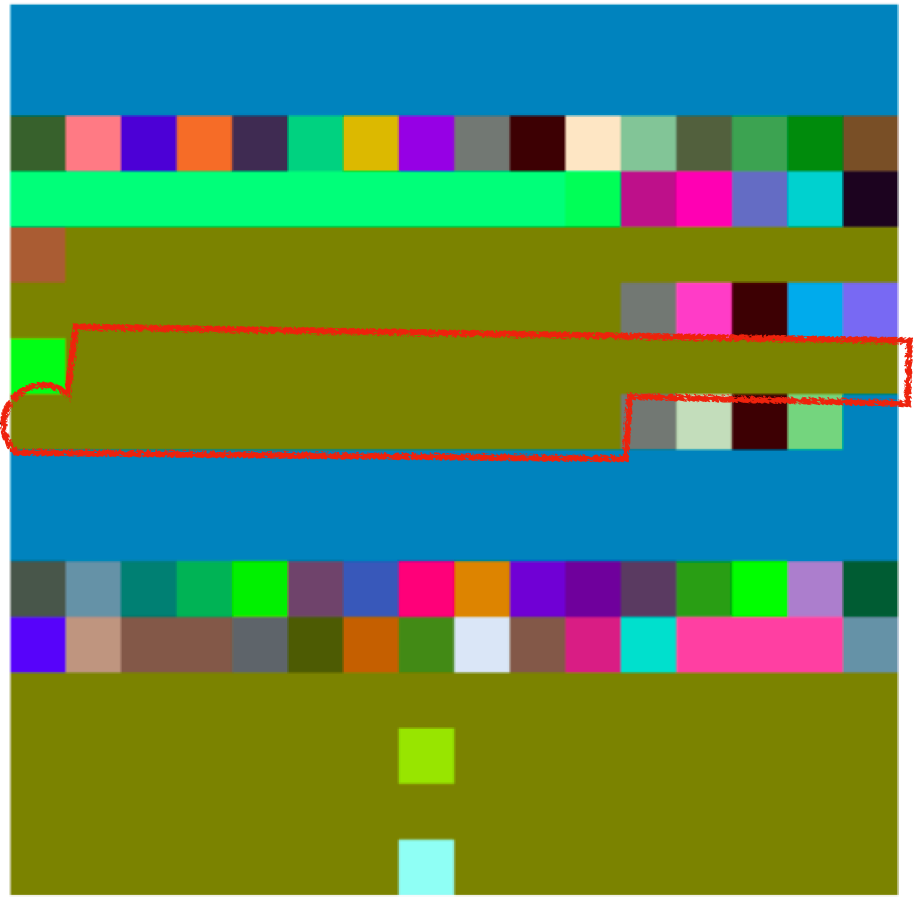
The first 256 characters

LATIN LETTER A - Z



The first 256 characters

LATIN LETTER a - z

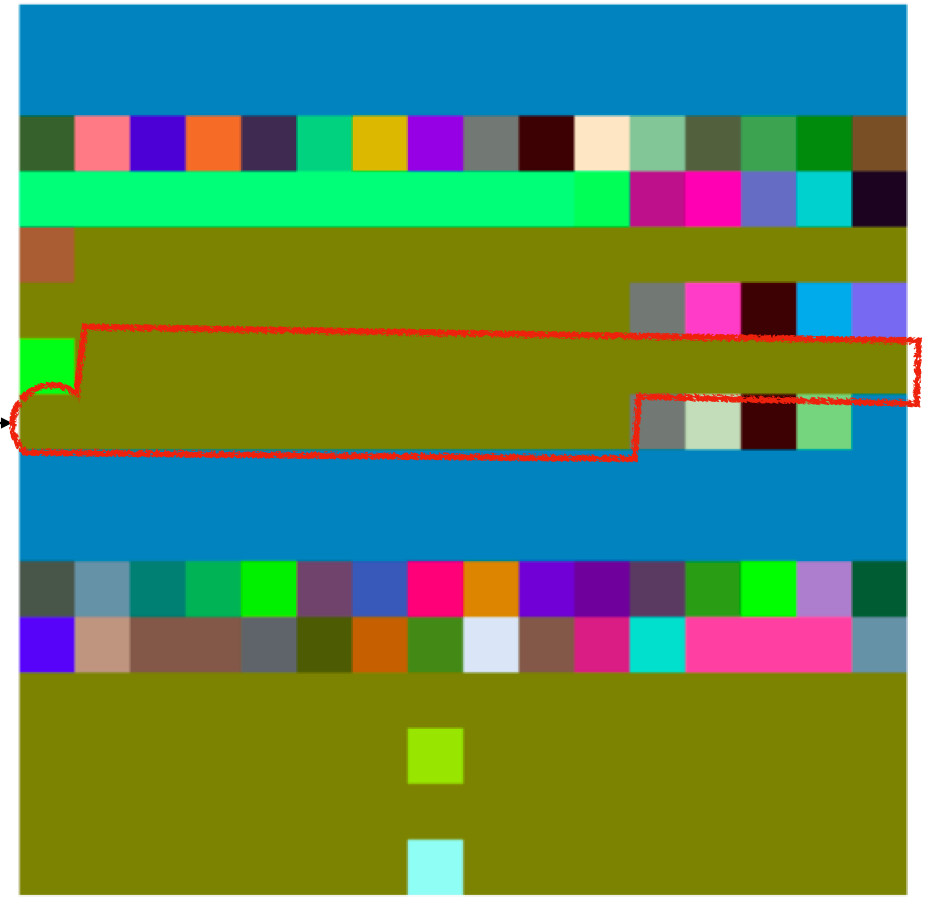


The first 256 characters

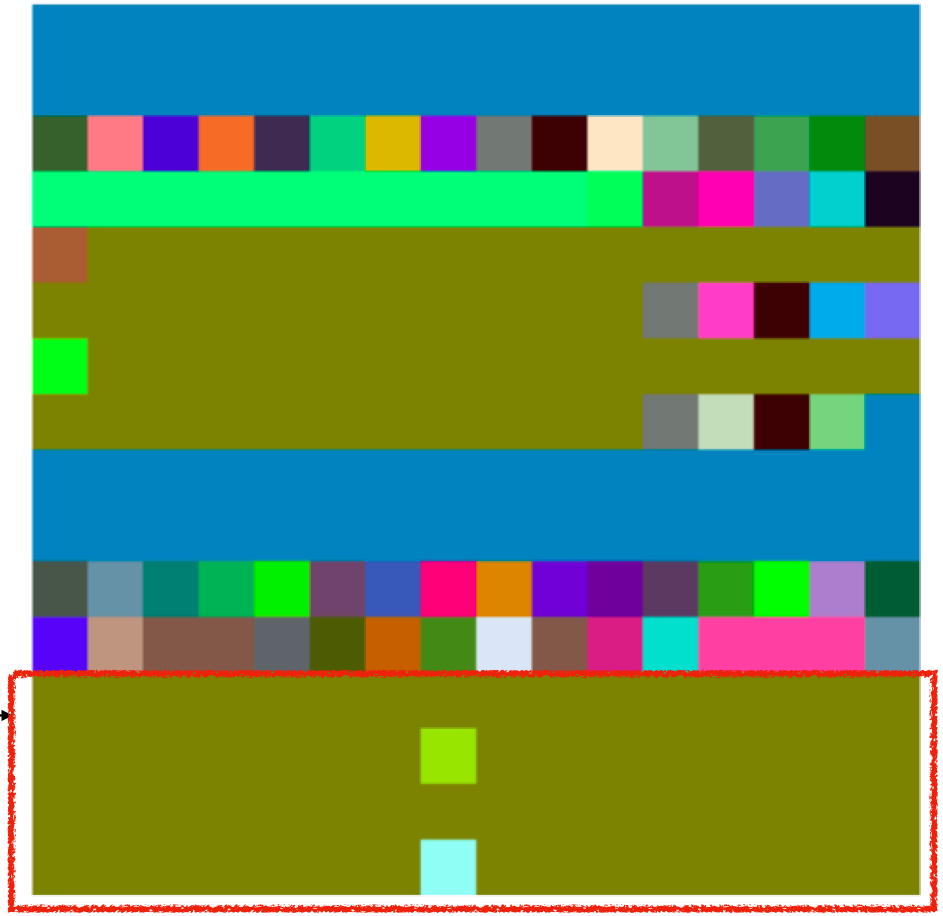
ASCII Code Chart

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
1	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2		!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	DEL

LATIN LETTER a - z



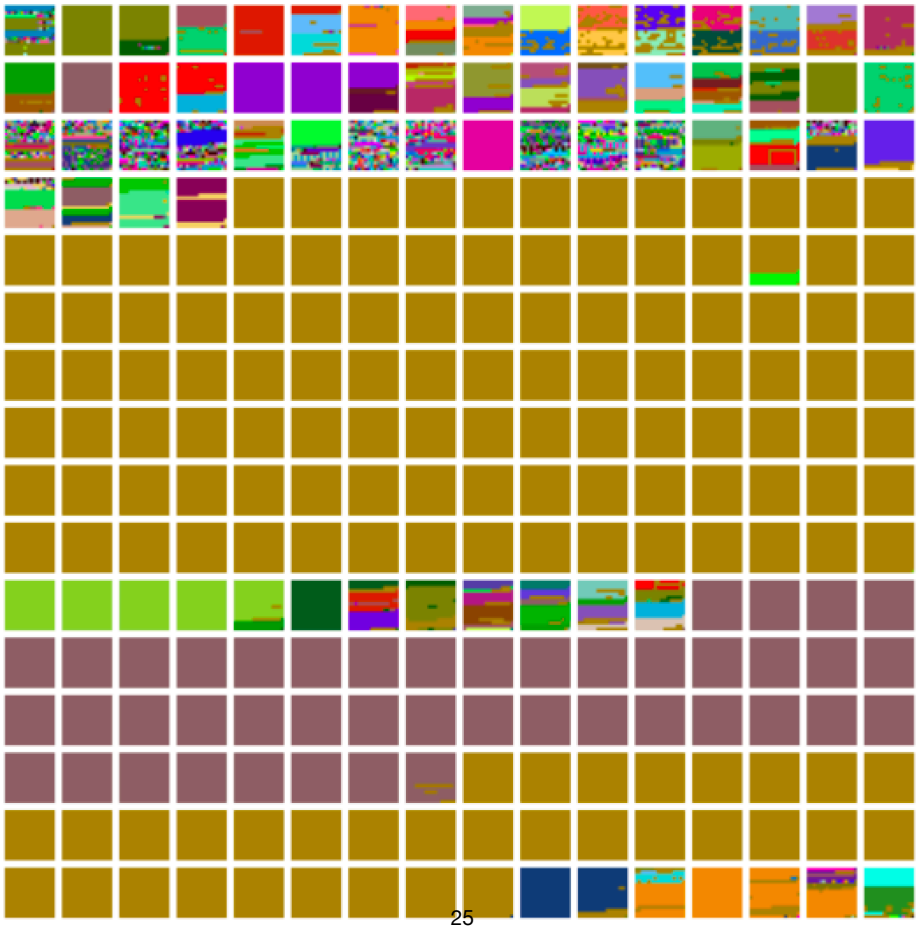
The first 256 characters



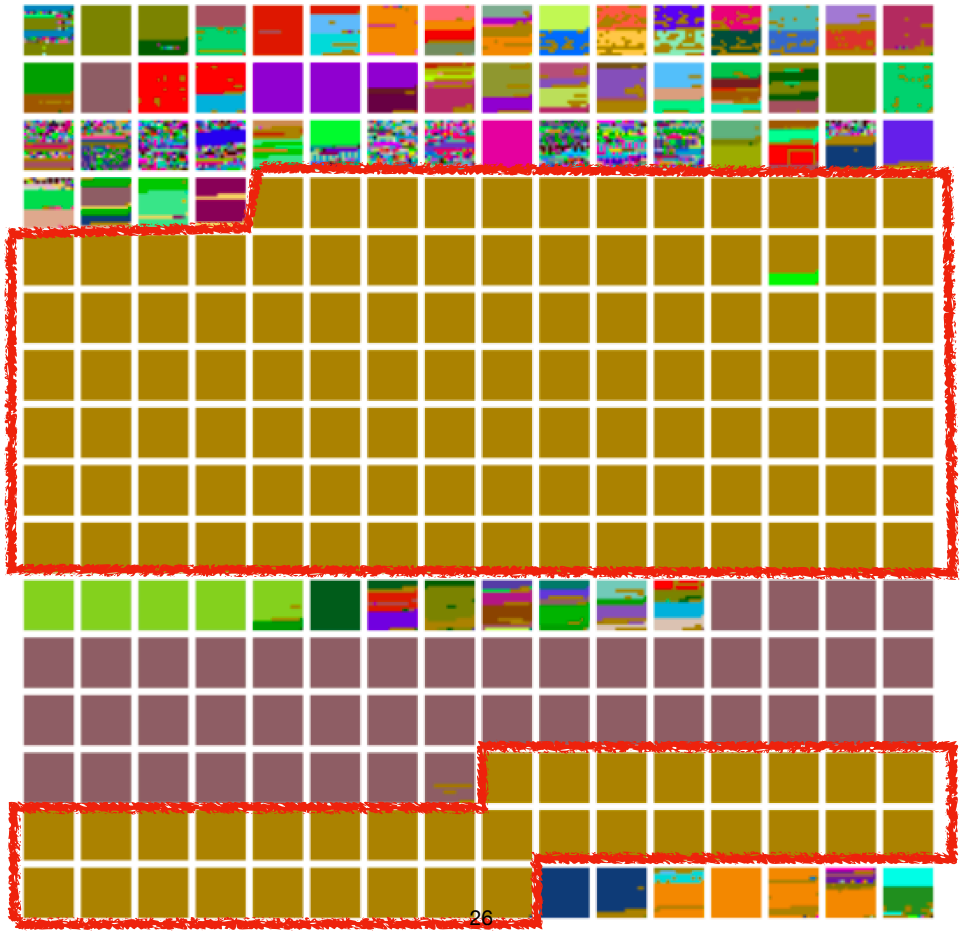
LATIN LETTER with

Accents: Ú ë í

The first 65K characters



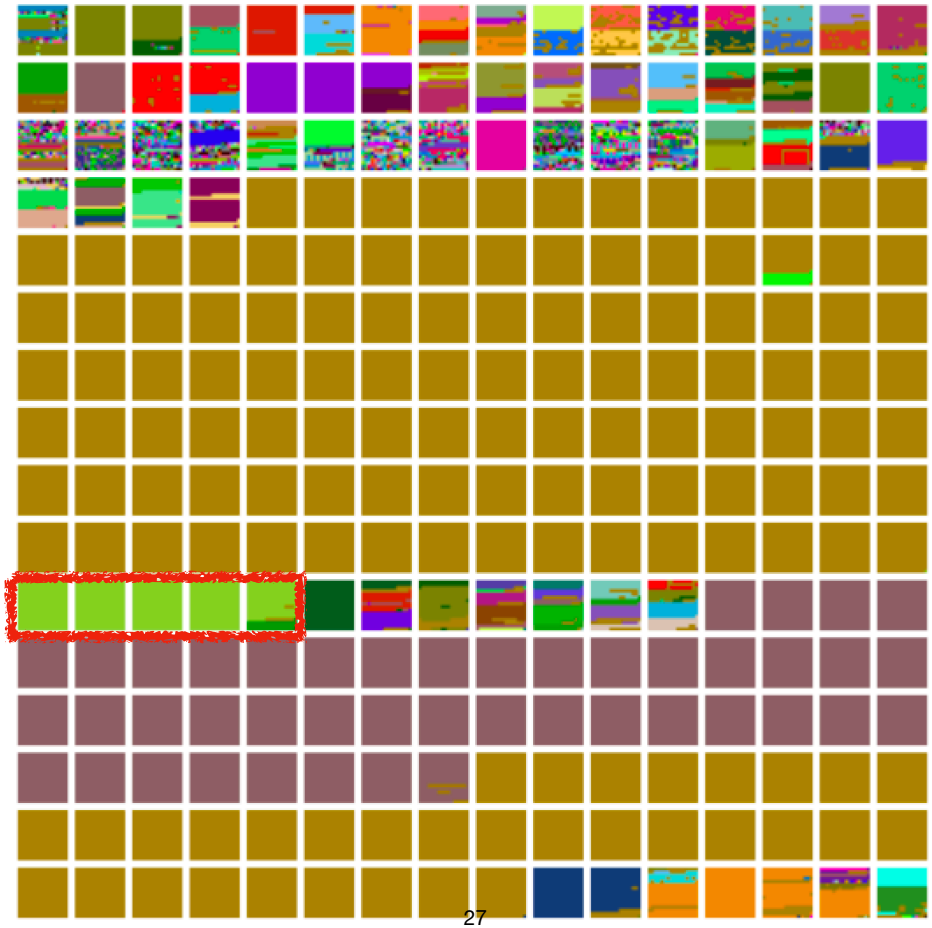
The first 65K characters



The first 65K characters

Yi

(ethnic group in China)



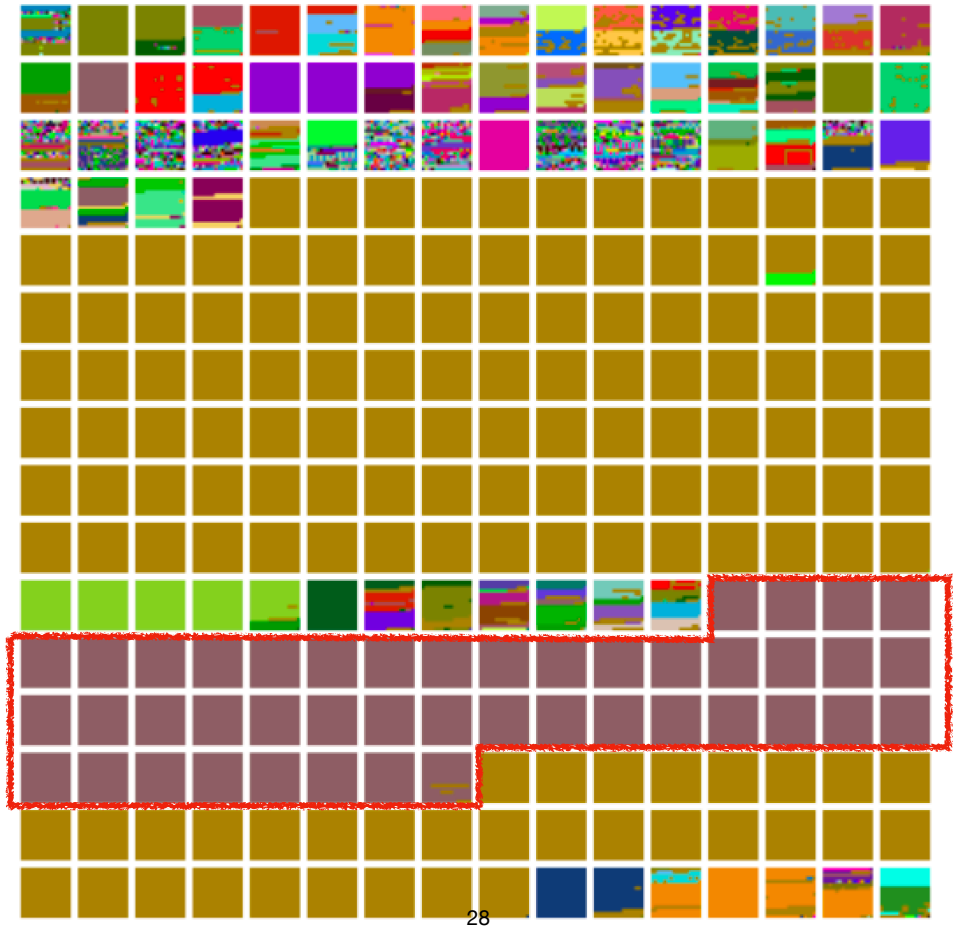
The first 65K characters

Hangul

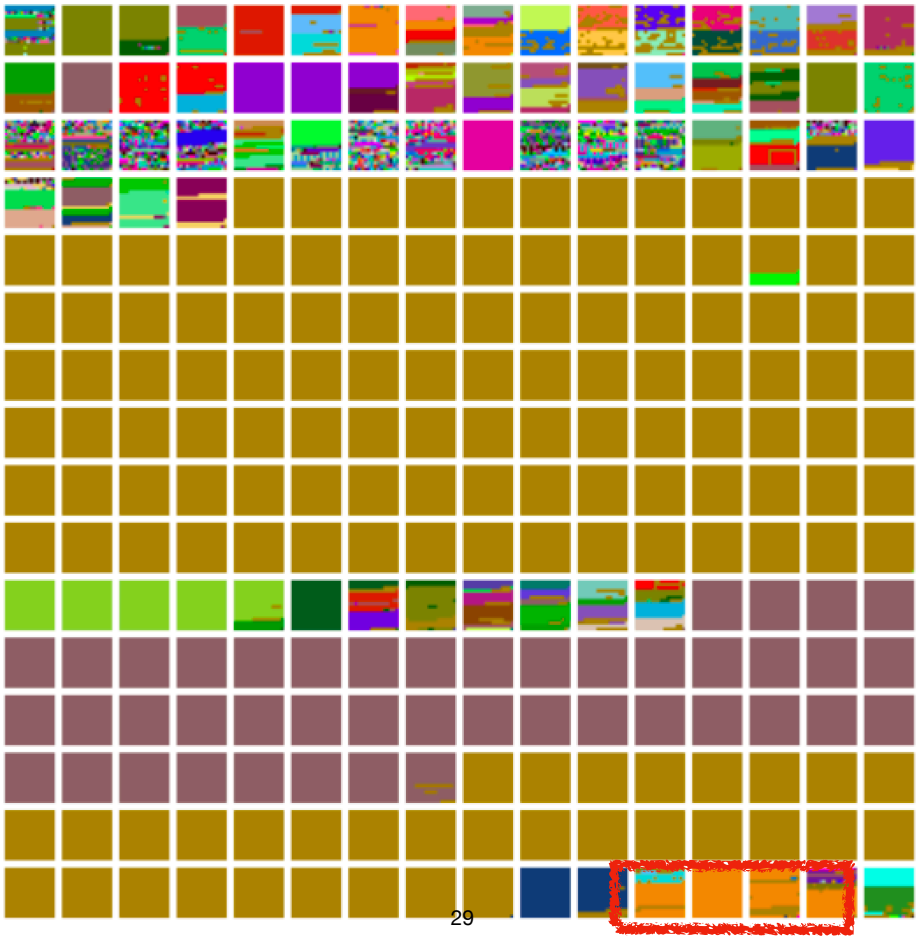
(Korean alphabet)

조선글

한글



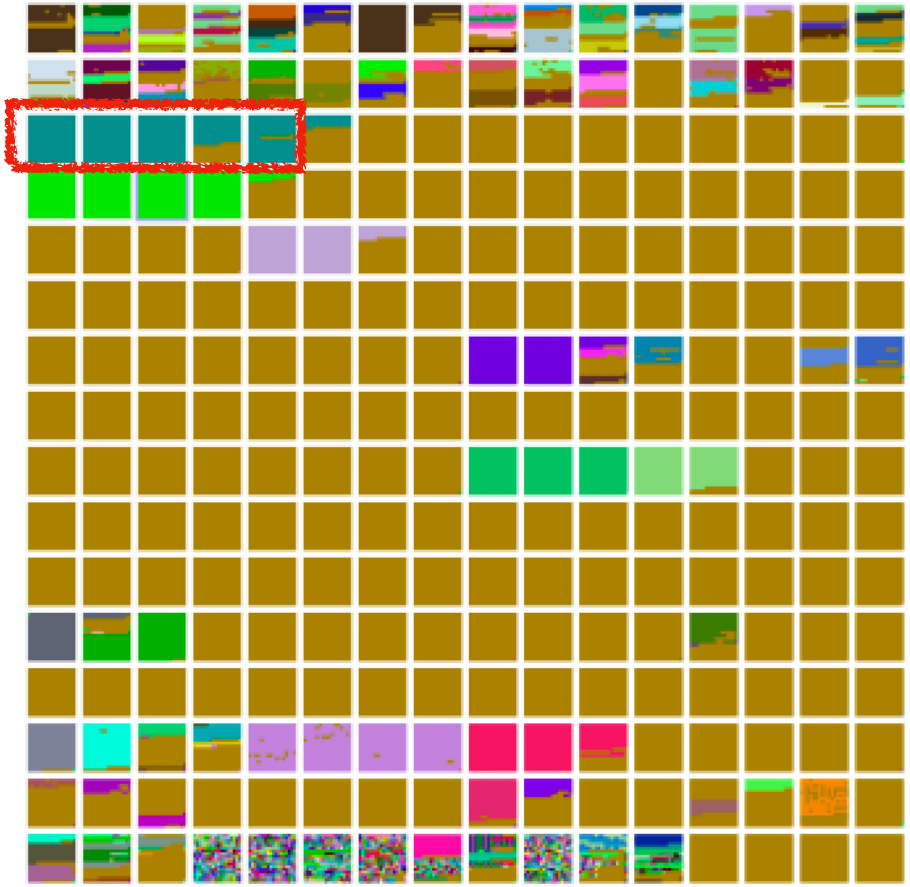
The first 65K characters



Arabic

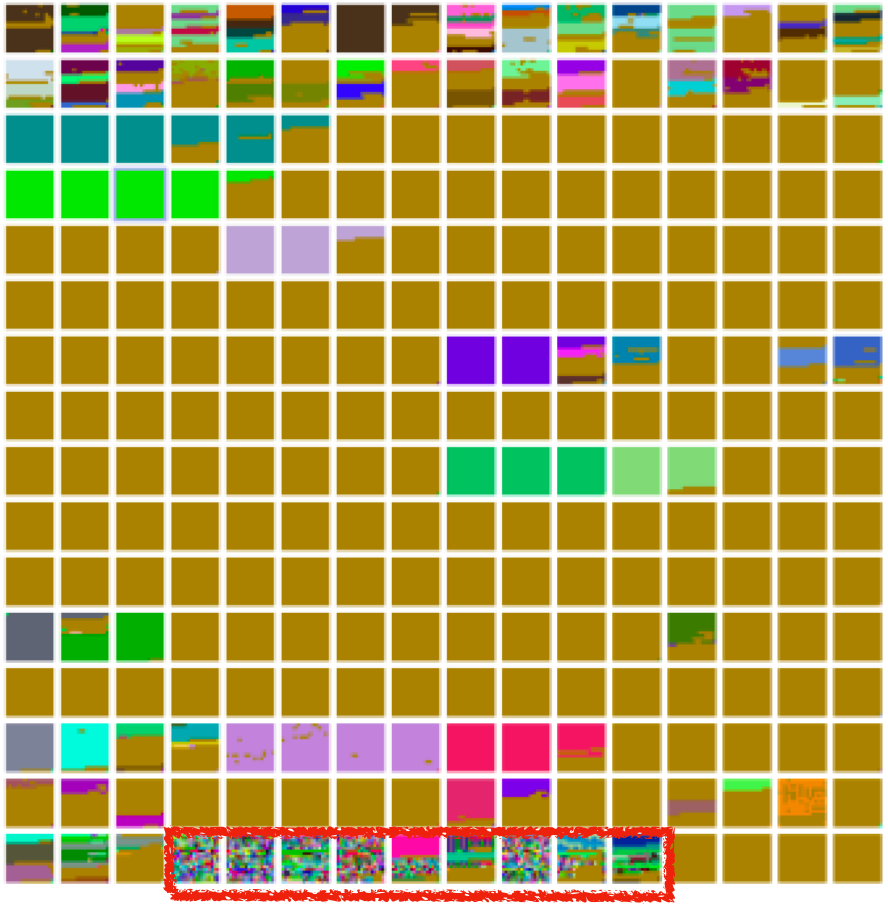
The second 65K characters

Cuneiform, invented by Sumerians in ancient Mesopotamia



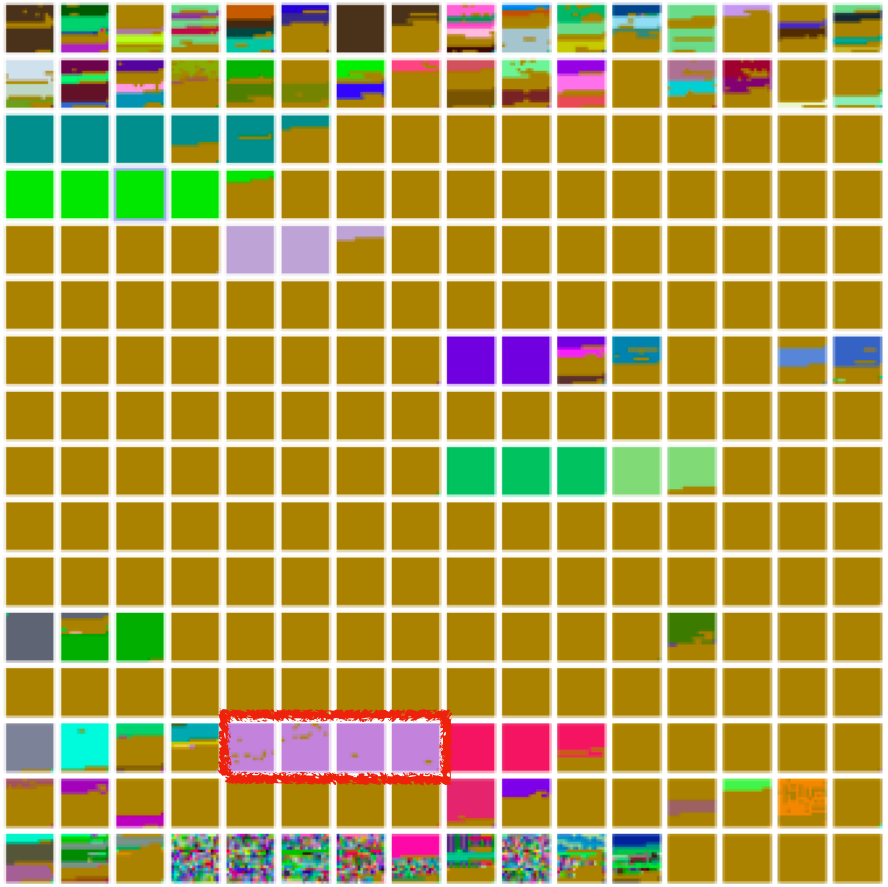
The second 65K characters

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
U+1F6Dx																
U+1F6Ex																
U+1F6Fx																
U+1F7Ex																
U+1F90x																
U+1F91x																
U+1F92x																
U+1F93x																
U+1F94x																
U+1F95x																
U+1F96x																



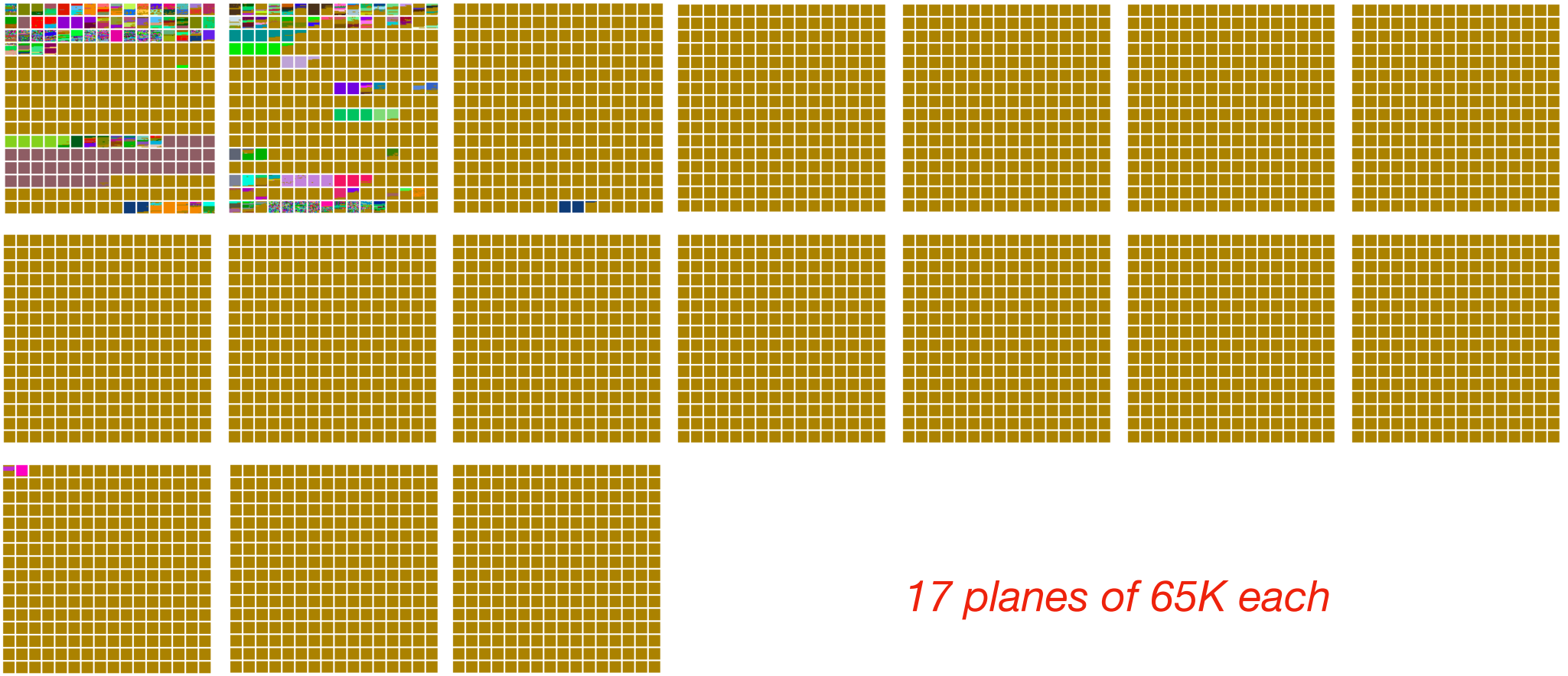
emoji

The second 65K characters



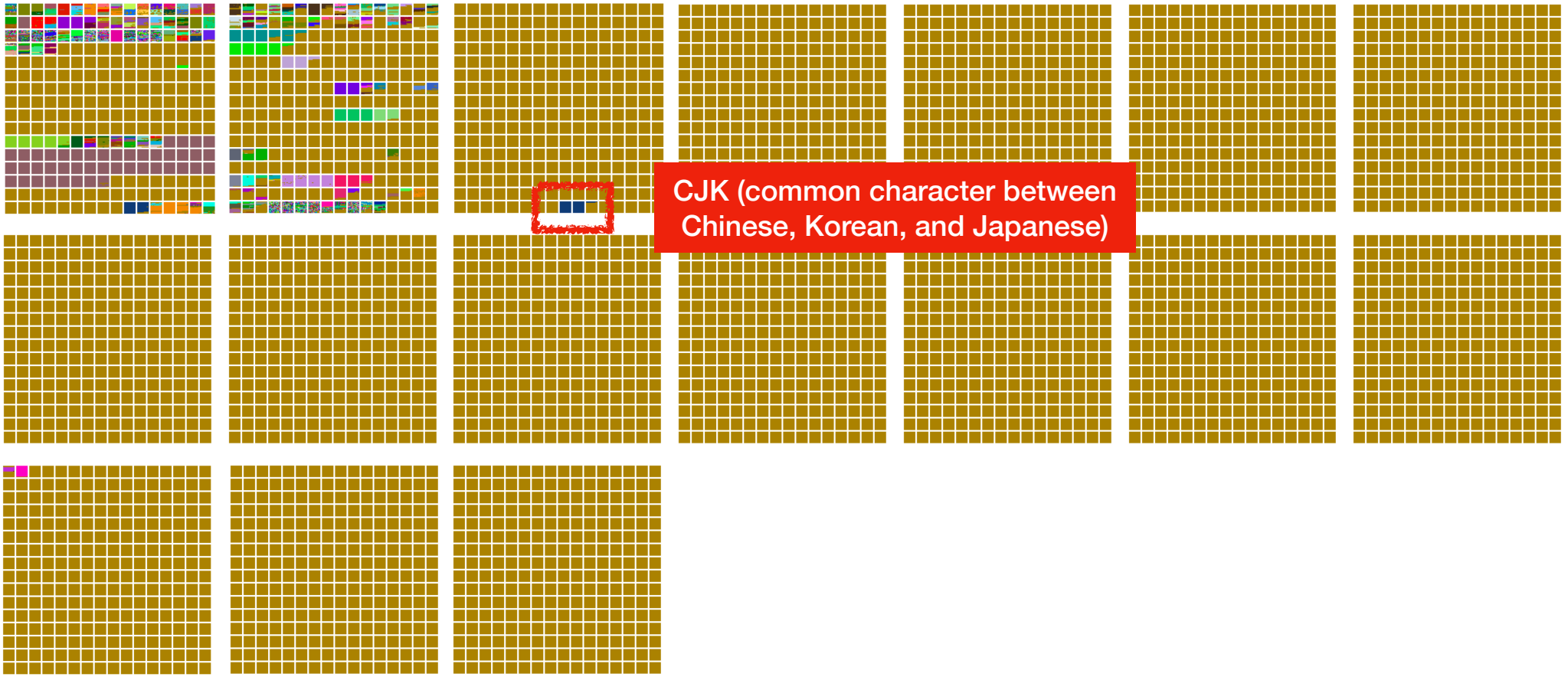
Mathematics

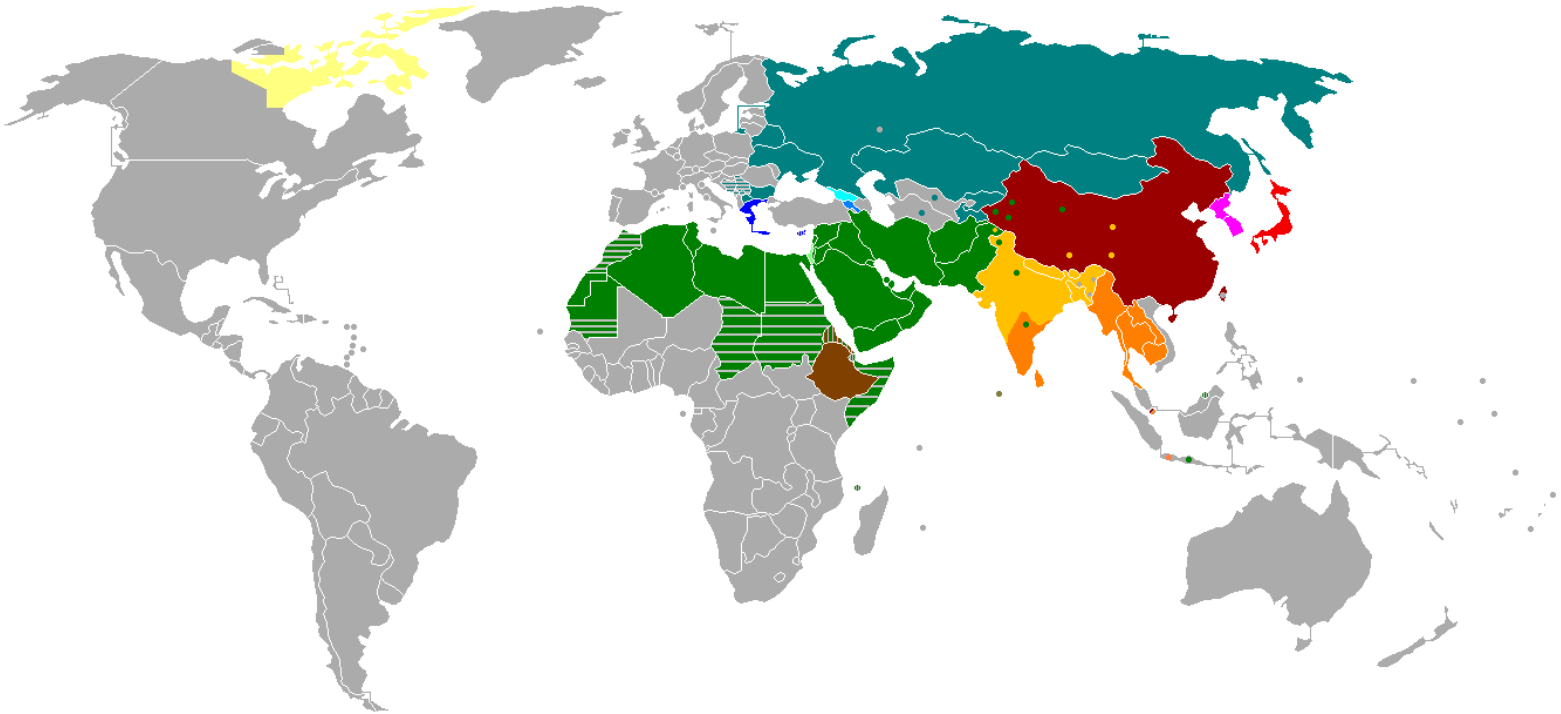
The 1 114 111 Unicode characters



17 planes of 65K each

The 1 114 111 Unicode characters





Index of predominant national and selected regional or minority scripts

Alphabetic	[L]ogographic and [S]yllabic	Abjad	Abugida
Latin	Hanzi [L]	Arabic	North Indic
Cyrillic	Kana [S] / Kanji [L]	Hebrew	South Indic
Greek	Hanja[b] [L]		Ethiopic
Armenian			Thaana
Georgian			Canadian syllabic
Hangul[a]			

Character encoding forms: UTFs (level 3/4)

- Representation of a scalar value in a computer
- No escape: a simple juxtaposition is a concatenation

Character latin A

- abstract character:
  - the letter A of the Latin script
- coded character:
  - name: LATIN CAPITAL LETTER A
  - code point: U+0041
- encoding forms:
  - UTF-8: 41
  - UTF-16: 0041

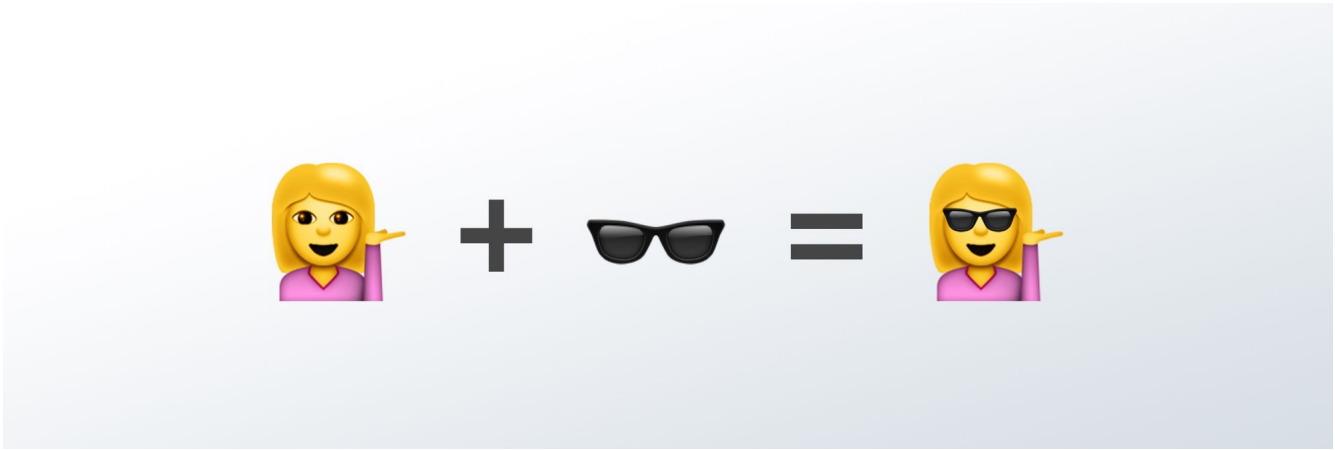
Character Hiragana MA

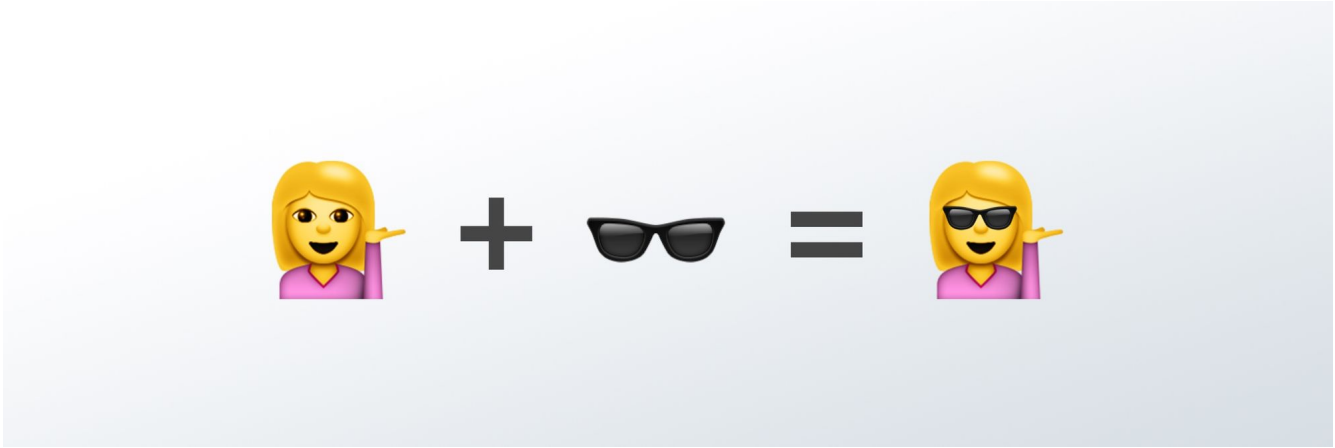
- abstract character:
  - the letter ま of the Hiragana script (Japanese, each made of 3 strokes)
- coded character:
  - name: HIRAGANA LETTER MA
  - code point: U+307E
- encoding forms:
  - UTF-8: E3 81 BE
  - UTF-16: 307E

Unicode encodes characters, not glyphs

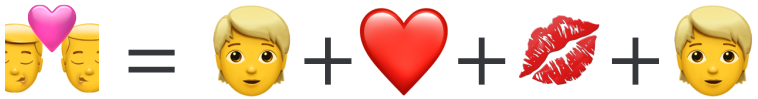
- The character U+0041 can equally well be displayed as A, **A**, *A*, ***A***, $\mathbb{A}$, …
- Sometimes different glyphs may be required
  - Egg in French is written œuf
- going from characters to glyphs: *shaping*

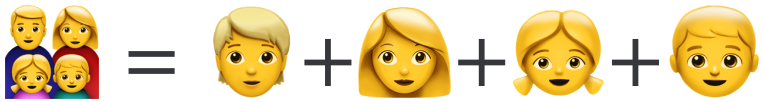
Zero Width Joiner (ZWJ)





The invisible glue character is called a Zero Width Joiner (ZWJ)[1] and a sequence of emojis joined together with a ZWJ character is known as an Emoji ZWJ Sequence.





👯 takes one character (i.e., it has a single code point)

👩‍❤️‍👩 includes three emojis but a total of six code points

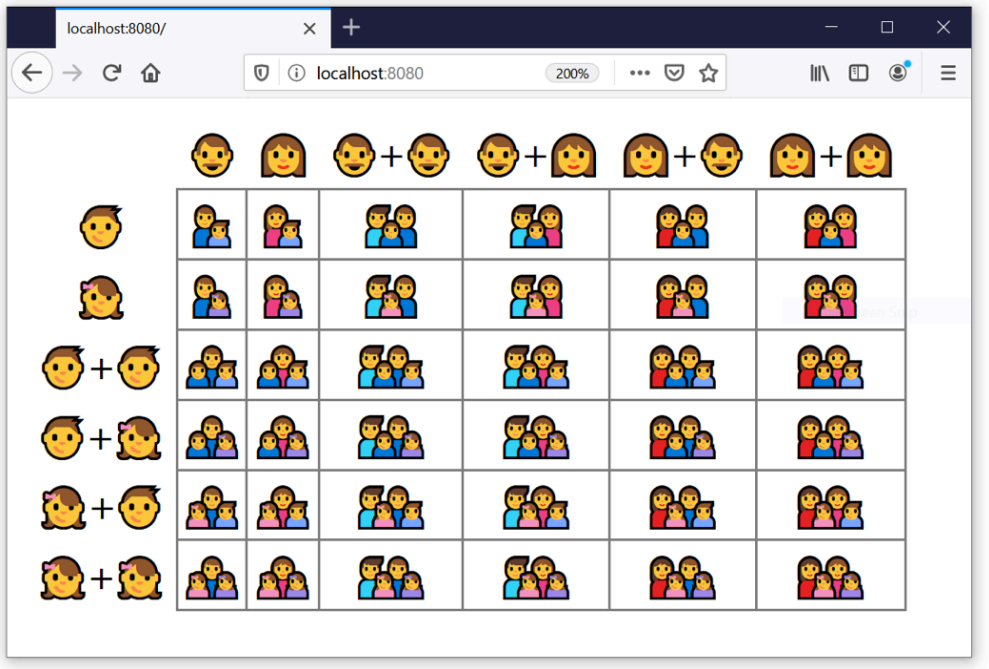
👯 takes one character (i.e., it has a single code point)

👩‍❤️‍👩 includes three emojis but a total of six code points

https://blog.emojipedia.org/emoji-zwj-sequences-three-letters-many-possibilities/

Building Multi-Character Emojis

Unicode has always supported building accented characters by combining letters and diacritics. This idea has been extended to meet the growing demand for emojis.



https://www.fluentpython.com/extra/multi-character-emojis/

Common practical issues

MULTIcaja

En alianza con PayPal

ï¿½Estï¿½s listo! Ya puedes comenzar a utilizar nuestro servicio de retiros

Hola Alexandre Bergel,

Comienza a transferir tus fondos en dï¿½lares desde PayPal a tu cuenta bancaria nacional en pesos chilenos.

Solicita un retiro de tus fondos PayPal:

1. Inicia sesiï¿½n en nuestro sitio web con tu RUT y Clave de 4 dï¿½gitos clic aquï¿½.
2. Ingresa el monto de tus fondos PayPal que quieres retirar.
3. Revisa el monto en dï¿½lares que vas a retirar, la tarifa y el monto en pesos que serï¿½ depositado en tu cuenta bancaria.
4. Confirma la solicitud del retiro.

Recibirï¿½s tu dinero en tu cuenta bancaria en Chile en 5 dï¿½as hï¿½biles.

Te saluda atentamente,

Multicaja en asociaciï¿½n con PayPal

Para asegurarte de recibir correctamente nuestros emails, agrega contacto@multicaja.cl a tu lista de direcciones.
Por favor no respondas este email. Si necesitas contactarte con nosotros llï¿½manos al 600 363 20 20

[redacted]

Re: CIEL 2014 : Demande d'Ã©valuation du papier N°6

To: Alexandre Bergel, Cc: [redacted]

Bonjour,

Oui, si cela vous est plus commode, pas de soucis.

- Christelle et Marie-Agnès

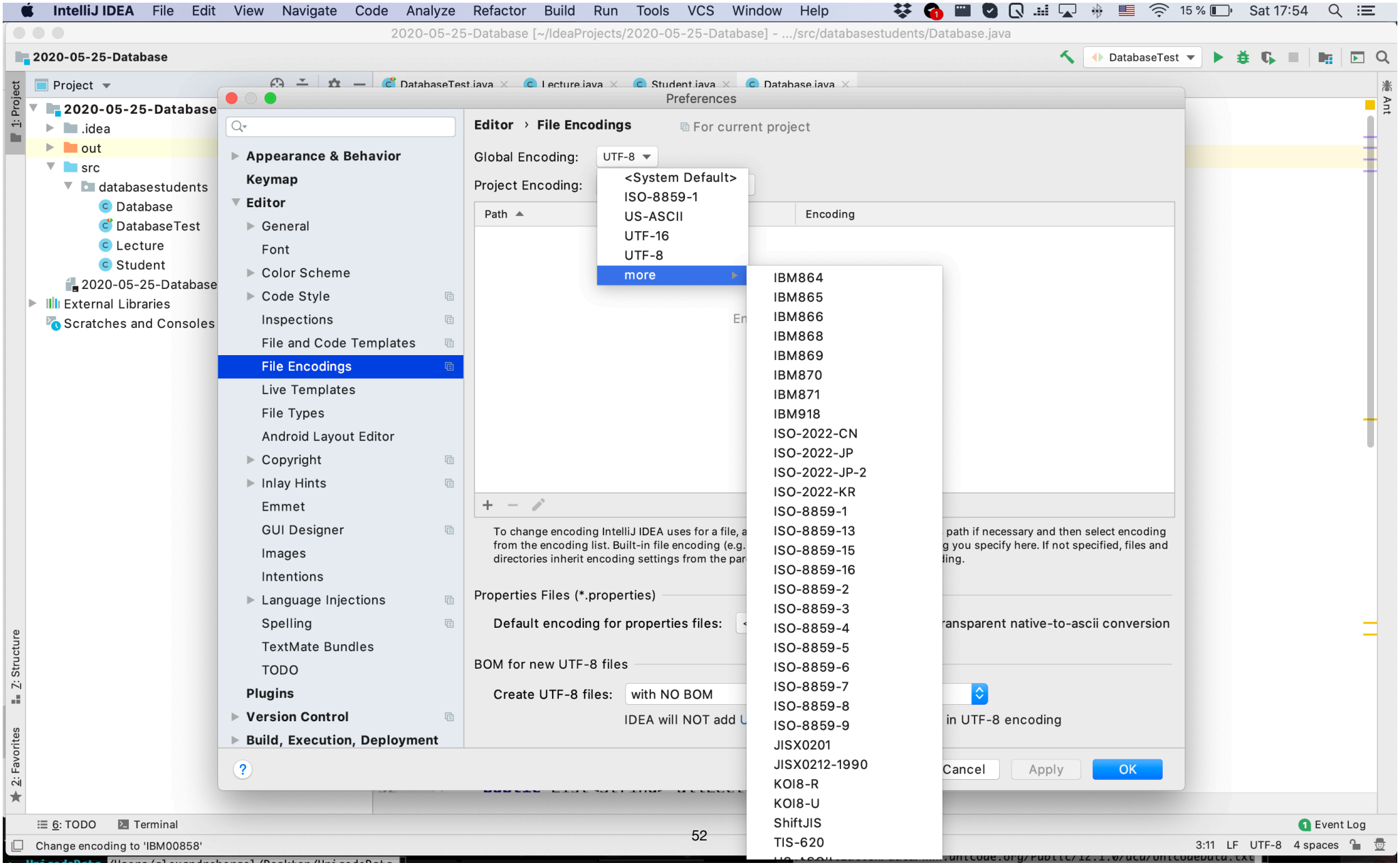
----- Mail original -----

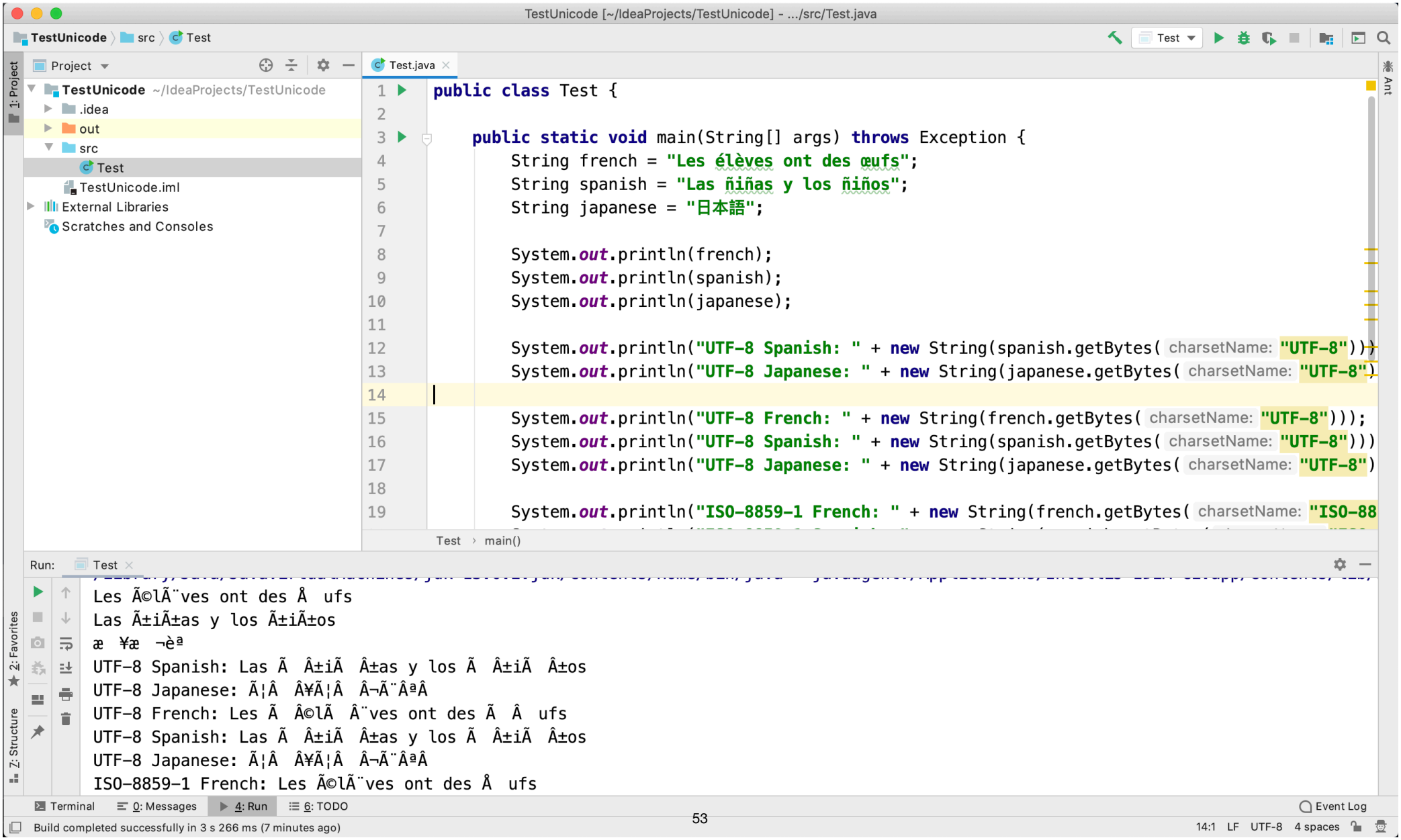
See More from Alexandre Bergel

```
public class Test {

    public static void main(String[] args) throws Exception {
        String french = "Les élèves ont des œufs";
        String spanish = "Las ñiñas y los ñiños";
        String japanese = "日本語";

        System.out.println(french);
        System.out.println(spanish);
        System.out.println(japanese);

        System.out.println("UTF-8 French: " + new String(french.getBytes("UTF-8")));
        System.out.println("UTF-8 Spanish: " + new String(spanish.getBytes("UTF-8")));
        System.out.println("UTF-8 Japanese: " + new String(japanese.getBytes("UTF-8")));

        System.out.println("ISO-8859-1 French: " + new String(french.getBytes("ISO-8859-1")));
        System.out.println("ISO-8859-1 Spanish: " + new String(spanish.getBytes("ISO-8859-1")));
        System.out.println("ISO-8859-1 japanese: " + new String(japanese.getBytes("ISO-8859-1")));
    }
}
```

```java
public class Test {

   public static void main(String[] args) throws Exception {
      String french = "Les élèves ont des œufs";
      String spanish = "Las ñiñas y los ñiños";
      String japanese = "日本語";

      System.out.println(french);
      System.out.println(spanish);
      System.out.println(japanese);

      System.out.println("UTF-8 French: " + new String(french.getBytes("UTF-8")));
      System.out.println("UTF-8 Spanish: " + new String(spanish.getBytes("UTF-8")));
      System.out.println("UTF-8 Japanese: " + new String(japanese.getBytes("UTF-8")));

      System.out.println("ISO-8859-1 French: " + new String(french.getBytes("ISO-8859-1")));
      System.out.println("ISO-8859-1 Spanish: " + new String(spanish.getBytes("ISO-8859-1")));
      System.out.println("ISO-8859-1 japanese: " + new String(japanese.getBytes("ISO-8859-1")));
   }
}
```

```
Les élèves ont des œufs
Las ñiñas y los ñiños
日本語
UTF-8 French: Les élèves ont des œufs
UTF-8 Spanish: Las ñiñas y los ñiños
UTF-8 Japanese: 日本語
ISO-8859-1 French: Les ?l?ves ont des ?ufs
ISO-8859-1 Spanish: Las ?i?as y los ?i?os
ISO-8859-1 japanese: ???
```


IntelliJ IDEA File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
2020-05-25-Database [~/IdeaProjects/2020-05-25-Database] - .../src/databasestudents/Database.java
Preferences
Editor › File Encodings
For current project
Global Encoding: UTF-8
Project Encoding:
Path
Encoding
<System Default>
ISO-8859-1
US-ASCII
UTF-16
UTF-8
more
IBM864
IBM865
IBM866
IBM868
IBM869
IBM870
IBM871
IBM918
ISO-2022-CN
ISO-2022-JP
ISO-2022-JP-2
ISO-2022-KR
ISO-8859-1
ISO-8859-13
ISO-8859-15
ISO-8859-16
ISO-8859-2
ISO-8859-3
ISO-8859-4
ISO-8859-5
ISO-8859-6
ISO-8859-7
ISO-8859-8
ISO-8859-9
JISX0201
JISX0212-1990
KOI8-R
KOI8-U
ShiftJIS
TIS-620
Properties Files (*.properties)
Default encoding for properties files:
BOM for new UTF-8 files
Create UTF-8 files:
with NO BOM
Cancel
Apply
OK
Change encoding to 'IBM00858'
3:11 LF UTF-8 4 spaces


TestUnicode [~/IdeaProjects/TestUnicode] - .../src/Test.java
TestUnicode
src
Test
Project
TestUnicode ~/IdeaProjects/TestUnicode
.idea
out
src
Test
TestUnicode.iml
External Libraries
Scratches and Consoles
Test.java
1 public class Test {
2
3 public static void main(String[] args) throws Exception {
4 String french = "Les élèves ont des œufs";
5 String spanish = "Las ñiñas y los ñiños";
6 String japanese = "日本語";
7
8 System.out.println(french);
9 System.out.println(spanish);
10 System.out.println(japanese);
11
12 System.out.println("UTF-8 Spanish: " + new String(spanish.getBytes(charsetName: "UTF-8")));
13 System.out.println("UTF-8 Japanese: " + new String(japanese.getBytes(charsetName: "UTF-8")
14
15 System.out.println("UTF-8 French: " + new String(french.getBytes(charsetName: "UTF-8")));
16 System.out.println("UTF-8 Spanish: " + new String(spanish.getBytes(charsetName: "UTF-8")))
17 System.out.println("UTF-8 Japanese: " + new String(japanese.getBytes(charsetName: "UTF-8")
18
19 System.out.println("ISO-8859-1 French: " + new String(french.getBytes(charsetName: "ISO-88
Test › main()
Run: Test
Les Ã©lÃ¨ves ont des Å ufs
Las Ã±iÃ±as y los Ã±iÃ±os
æ ¥æ ¬èª
UTF-8 Spanish: Las Ã Â±iÃ Â±as y los Ã Â±iÃ Â±os
UTF-8 Japanese: Ã¦Â Â¥Ã¦Â Â¬Ã¨Âª
UTF-8 French: Les Ã Â©lÃ Â¨ves ont des Ã Â ufs
UTF-8 Spanish: Las Ã Â±iÃ Â±as y los Ã Â±iÃ Â±os
UTF-8 Japanese: Ã¦Â Â¥Ã¦Â Â¬Ã¨Âª
ISO-8859-1 French: Les Ã©lÃ¨ves ont des Å ufs
Terminal
0: Messages
4: Run
6: TODO
Event Log
Build completed successfully in 3 s 266 ms (7 minutes ago)
14:1 LF UTF-8 4 spaces
1: Project
2: Favorites
7: Structure
Ant

Useful tips

- Never, ever use Windows' Notepad to write code
  - It uses ISO 8859-1, which leads to numerous problems
- *Use UTF-8* all the way down:
  - For the text files used in your program
  - For the source code used in your program
- Make sure that your programming environment is set to UTF-8
- Always specify UTF-8 when *opening for reading or writing a textual file*

Useful tips

Text files:

```
Reader reader = new InputStreamReader(new FileInputStream("/tmp/foo.txt", "UTF-8"));
Writer writer = new OutputStreamWriter(new FileOutputStream("/tmp/foo.txt", "UTF-8"));
```

String files:

```
byte[] bytesInDefaultEncoding = someString.getBytes(); // May generate corrupt bytes.
byte[] bytesInUTF8 = someString.getBytes("UTF-8"); // Correct.
String stringUsingDefaultEncoding = new String(bytesInUTF8); // Unknown bytes becomes "?".
String stringUsingUTF8 = new String(bytesInUTF8, "UTF-8"); // Correct.
```

Fun

```
➜  T cat Unicode.java
\u0070\u0075\u0062\u006C\u0069\u0063\u0020\u0020\u0020\u0063\u006C\u0061\u0073\u0073\u0020\u0020
\u0055\u006E\u0069\u0063\u006F\u0064\u0065\u0020\u007B\u0020\u0070\u0075\u0062\u006C\u0069\u0063
\u0020\u0020\u0073\u0074\u0061\u0074\u0069\u0063\u0020\u0020\u0076\u006F\u0069\u0064\u0020\u0020
\u006D\u0061\u0069\u006E\u0020\u0028\u0020\u0053\u0074\u0072\u0069\u006E\u0067\u0020\u005B\u005D
\u0061\u0072\u0067\u0073\u0020\u0029\u0020\u007B\u0020\u0053\u0079\u0073\u0074\u0065\u006D\u002E
\u006F\u0075\u0074\u002E\u0070\u0072\u0069\u006E\u0074\u006C\u006E\u0028\u0022\u0049\u0022\u002B
\u0022\u0020\u2665\u0020\u0055\u006E\u0069\u0063\u006F\u0064\u0065\u0022\u0029\u003B\u007D\u007D
➜  T javac Unicode.java
➜  T java Unicode
I ♥ Unicode
```

\u0070\u0075\u0062\u006C\u0069\u0063\u0020\u0020\u0020\u0063\u006C\u0061\u0073\u0073\u0020\u0020
\u0055\u006E\u0069\u0063\u006F\u0064\u0065\u0020\u007B\u0020\u0070\u0075\u0062\u006C\u0069\u0063
\u0020\u0020\u0073\u0074\u0061\u0074\u0069\u0063\u0020\u0020\u0076\u006F\u0069\u0064\u0020\u0020
\u006D\u0061\u0069\u006E\u0020\u0028\u0020\u0053\u0074\u0072\u0069\u006E\u0067\u0020\u005B\u005D
\u0061\u0072\u0067\u0073\u0020\u0029\u0020\u007B\u0020\u0053\u0079\u0073\u0074\u0065\u006D\u002E
\u006F\u0075\u0074\u002E\u0070\u0072\u0069\u006E\u0074\u006C\u006E\u0028\u0022\u0049\u0022\u002B
\u0022\u0020\u2665\u0020\u0055\u006E\u0069\u0063\u006F\u0064\u0065\u0022\u0029\u003B\u007D\u007D

The Unicode Consortium

Mission

- “making the digital world more inclusive”
- “Everyone in the world should be able to use their own language on phones and computers.”

Membership

- The effort of the Unicode Consortium is sponsored by membership fees
- Anyone can be a membership (student = 35 USD, corporation up to 21K USD)
- A member
  - has access to the technical discussions
  - can vote on many aspects of the consortium
- Actual members are large IT companies and some governments

Adopt a character

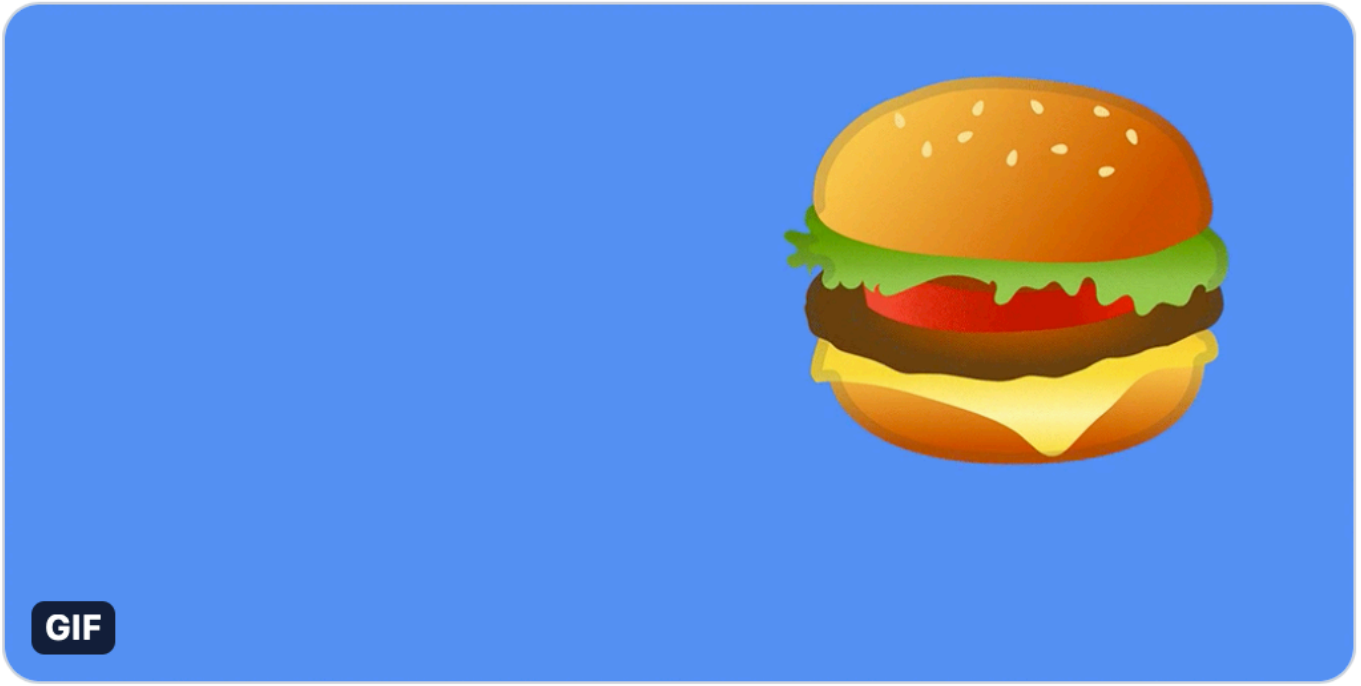
- Do you want your company to be associated with the  emoji?
- Do you want to declare your love with a  or  dedication for your partner?

54 Gold Sponsors

{ Elastic	} Elastic	& Adobe Inc.
Catalyst IT Limited	ɑ Ann Lewnes and Greg Welch	Suzi Slavik
台 Yaya Wang	愛 Enjoy the present（致独一无二的莎莎）	Cal Henderson
Jason Jenkins	Digital Love Media	discourse.org
Vinton G. Cerf	In loving memory of Ryan Neale Cross (1984-2017)	R. Martin Chavez & Adam Norbury
White Unicorn Agency	Oakland Athletics	Michael D'Errico
Norton Pedee Family	Indian Type Foundry	Candice DeStefano
Oakland Athletics	USA Pears	S&A Group
Zespri Kiwifruit	Avocados From Mexico	Scott Spears
Richard Shupak	Buffalo Wild Wings	Google

Google Design @GoogleDesign · Feb 14, 2018

No matter where the cheese goes, now it's official → #GoogleFonts adopted the @unicode #EmojiBurger on behalf of @Google! 💙🍔



8 35 132

Final words

Unicode

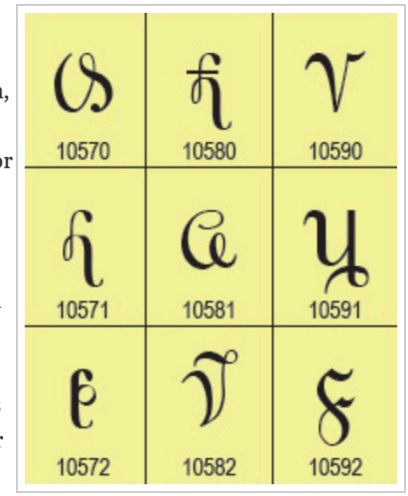
- *Awesome environment* to learn about a central aspect to many other cultures
- UTF-8, UTF-8, UTF-8, UTF-8, UTF-8, UTF-8, UTF-8, UTF-8, …
- Make sure you use
  - a proper professional programming environment and
  - text editing tool

UNICODE

NEWS, ANNOUNCEMENTS, RELEASE INFO, AND CALENDAR UPDATES FROM THE UNICODE CONSORTIUM

TUESDAY, SEPTEMBER 14, 2021

Announcing The Unicode® Standard, Version 14.0



Version 14.0 of the Unicode Standard is now available, including the core specification, annexes, and data files. This version adds 838 characters, for a total of 144,697 characters. These additions include five new scripts, for a total of 159 scripts, as well as 37 new emoji characters.

The new scripts and characters in Version 14.0 add support for modern language groups in Bosnia, India, Indonesia, Iran, Java, Malaysia, Mongolia, Myanmar, Pakistan, and the Philippines, plus other languages in Africa and North America, including:

- Arabic script additions that include honorifics and additions for Quranic use, and characters used to write languages across Africa, the Balkans, and South and Southeast Asia

LINKS OF INTEREST

What is Unicode?
The Unicode Consortium
Archived Announcements

BLOG ARCHIVE

▼ 2021 (27)
- ► November (2)
- ► October (3)
- ▼ September (3)
  - Announcing The Unicode® Standard, Version 14.0
  - Unicode CLDR v40 Alpha available for testing
  - Unicode Consortium Announces Version 14.0 Cover De...
- ► August (1)
- ► July (2)
- ► June (1)

http://blog.unicode.org/2021/09/announcing-unicode-standard-version-140.html

References

- https://www.unicode.org
- http://reedbeta.com/blog/programmers-intro-to-unicode/
- https://balusc.omnifaces.org/2009/05/unicode-how-to-get-characters-right.html
- https://www.joelonsoftware.com/2003/10/08/the-absolute-minimum-every-software-developer-absolutely-positively-must-know-about-unicode-and-character-sets-no-excuses/
- https://www.joelonsoftware.com/2003/10/08/the-absolute-minimum-every-software-developer-absolutely-positively-must-know-about-unicode-and-character-sets-no-excuses/
- https://en.wikipedia.org/wiki/Mojibake
- https://www.unicode.org/notes/tn23/Muller-Slides+Narr.pdf
- http://www.unicode.org/versions/Unicode13.0.0/ch02.pdf#G14527

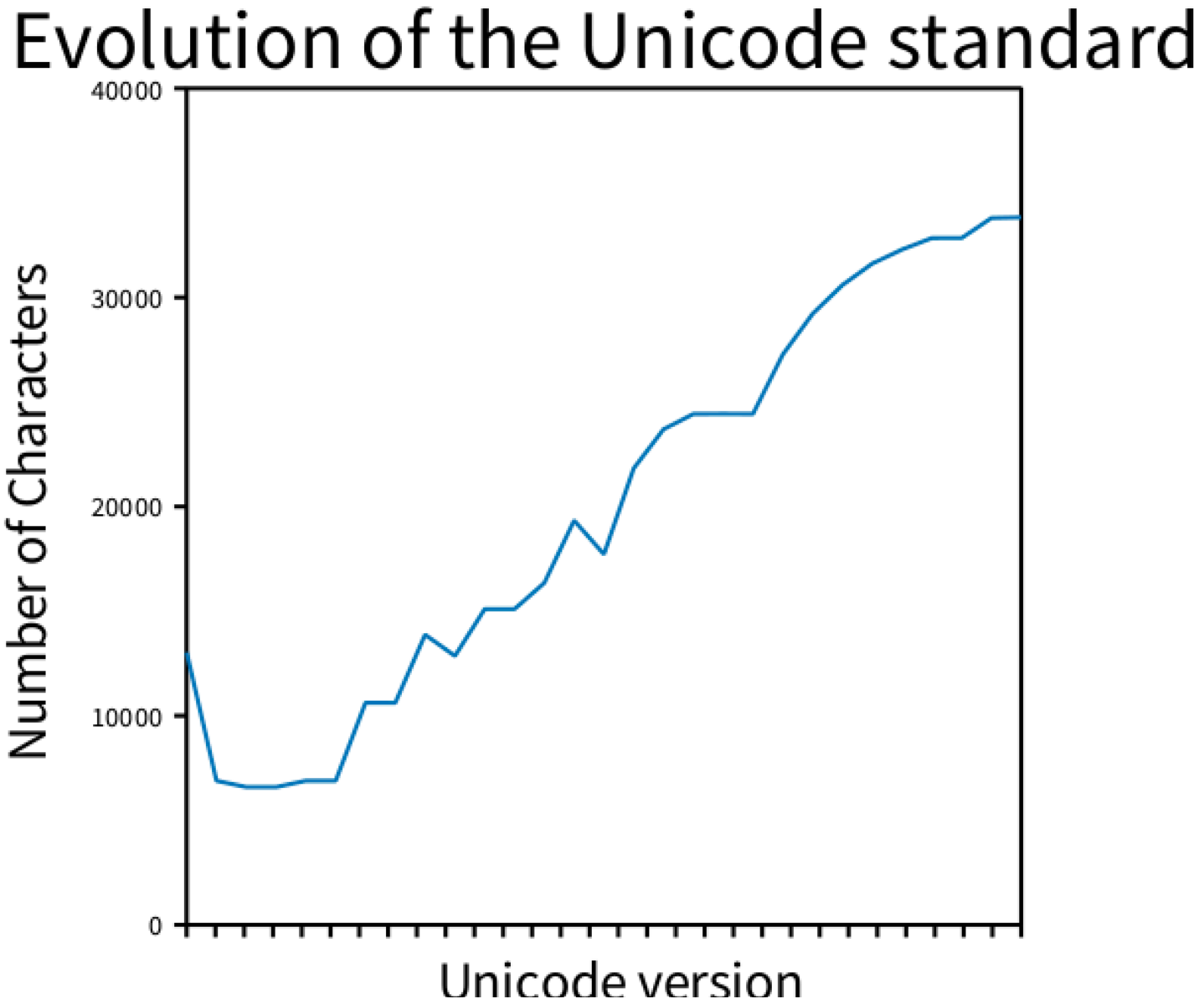
Introduction To Unicode

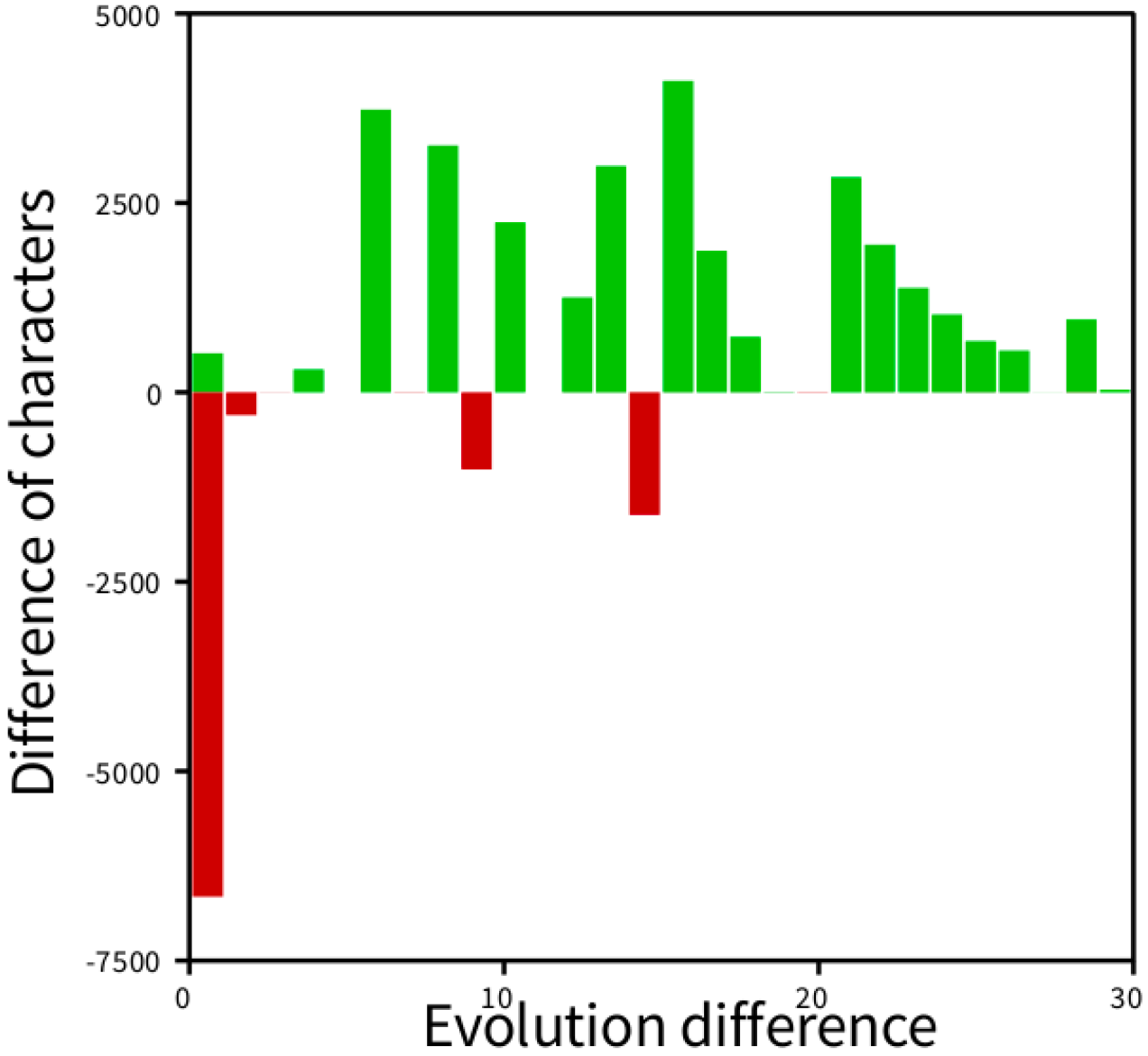
Making the digital world more inclusive

Alexandre Bergel
Computer Science Department - FCFM
University of Chile
http://bergel.eu **abergel@dcc.uchile.cl** **@AlexBergel**

Overview

- 29 publicly available version
- Version 1.1.5 established in 1995
- Version 14.0.0 defined in March 2020


Evolution of the Unicode standard
40000
30000
20000
10000
0
Number of Characters
Unicode version


5000
2500
0
-2500
-5000
-7500
Difference of characters
0
10
20
30
Evolution difference