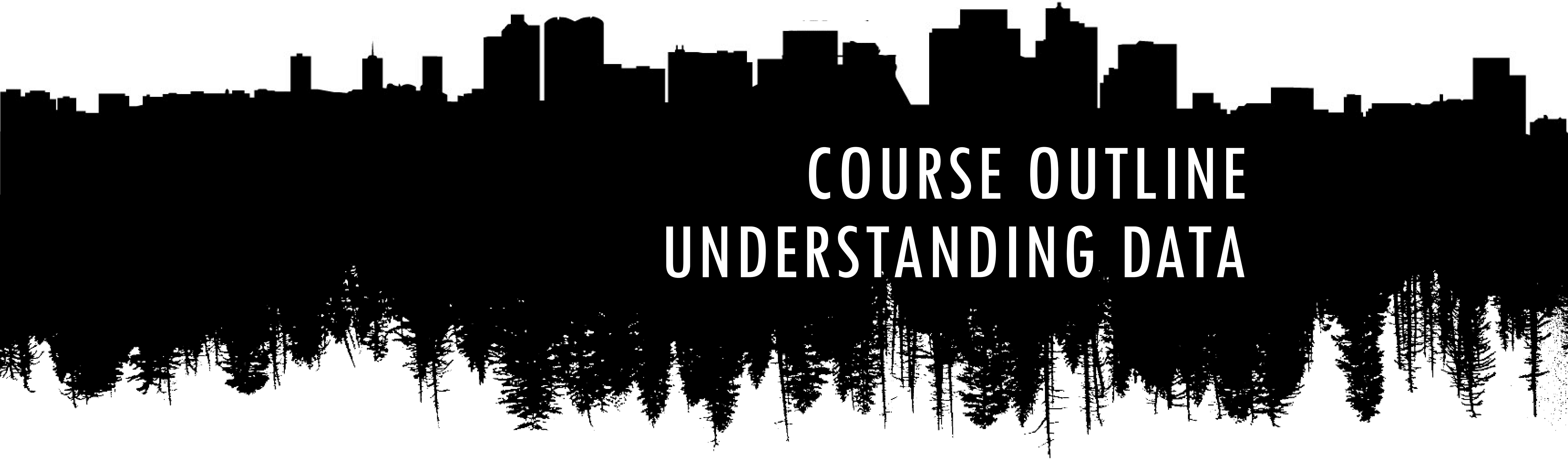
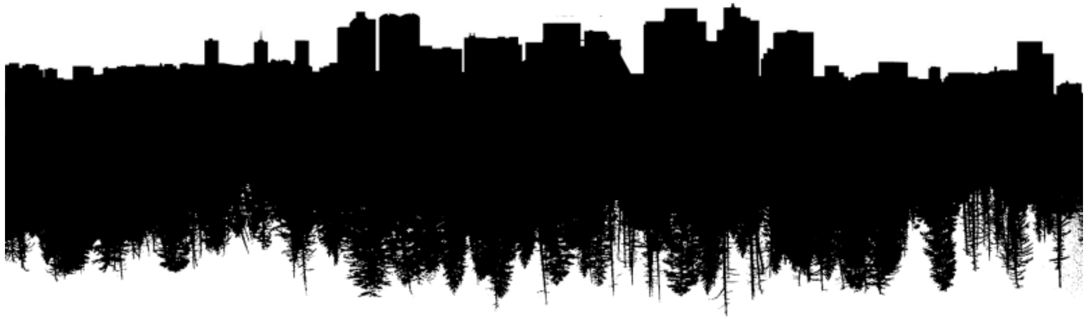


COURSE OUTLINE
UNDERSTANDING DATA

URBAN STUDIES AND PLANNING
DIGITAL URBAN
MONDAY 17TH JANUARY 9:00-12:00

Anssi Joutsiniemi
D.Sc(Tech), Architect
Professor of Practice
Aalto University



COURSE OUTLINE

<USP-E0363> COURSE IN NUTSHELL

COURSE INFORMATION:

MyCourses:

https://mycourses.aalto.fi/course/view.php?id=32810

ONLINE LECTURES:

Zoom:

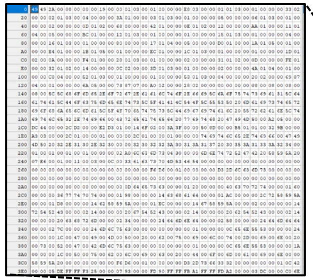
https://aalto.zoom.us/j/67921221869?pwd=cG5NYml2elh1alhiYmsyZDNKbUR0QT09

COURSE OUTCOME:

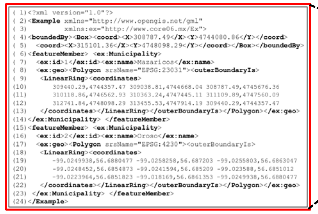
Your DATA FACORY:

One single notebook at notebooks.csc.fi/

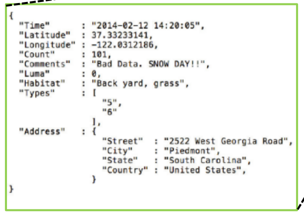
GENERALIZED DATA FACTORY DIAGRAM



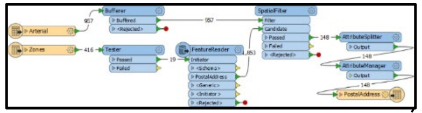
```
  0 49 49 2A 00 08 00 00 00 19 00 00 01 03 00 01 00 00 00 E8 03 00 00 01 01 03 00 01 00 00 00 33 02
 20 00 00 02 01 03 00 04 00 00 00 3A 01 00 00 03 01 03 00 01 00 00 00 05 00 00 00 06 01 03 00 01 00
 40 00 00 02 00 00 00 0D 01 02 00 68 00 00 00 42 01 00 00 0E 01 02 00 12 00 00 00 AA 01 00 00 11 01
 60 04 00 05 00 00 00 BC 01 00 00 12 01 03 00 01 00 00 00 01 00 00 00 15 01 03 00 01 00 00 00 04 00
 80 00 00 16 01 03 00 01 00 00 00 80 00 00 00 17 01 04 00 05 00 00 00 D0 01 00 00 1A 01 05 00 01 00
 A0 00 00 E4 01 00 00 1B 01 05 00 01 00 00 00 EC 01 00 00 1C 01 03 00 01 00 00 00 01 00 00 00 1D 01
 C0 02 00 0A 00 00 00 F4 01 00 00 28 01 03 00 01 00 00 00 02 00 00 00 31 01 02 00 0D 00 00 00 FE 01
 E0 00 00 32 01 02 00 14 00 00 00 0C 02 00 00 3D 01 03 00 01 00 00 00 02 00 00 00 4A 01 04 00 01 00
100 00 00 C8 04 00 00 52 01 03 00 01 00 00 00 01 00 00 00 53 01 03 00 04 00 00 00 20 02 00 00 69 87
120 04 00 01 00 00 00 6A 05 00 00 73 87 07 00 A0 02 00 00 28 02 00 00 00 00 00 00 08 00 08 00 08 00
140 08 00 5C 5C 68 6F 6D 65 2E 6F 72 67 2E 61 61 6C 74 6F 2E 66 69 5C 6A 6F 75 74 73 69 61 31 5C 64
160 61 74 61 5C 44 6F 63 75 6D 65 6E 74 73 5C 5F 41 41 4C 54 4F 5C 55 53 50 20 6D 61 69 73 74 65 72
180 69 6F 68 6A 65 6C 6D 61 5C 5F 4F 70 65 74 75 73 5C 44 69 67 69 74 61 6C 20 55 72 62 61 6E 5C 74
1A0 69 74 6C 65 32 2E 74 69 66 00 43 72 65 61 74 65 64 20 77 69 74 68 20 47 49 4D 50 00 A2 05 00 00
1C0 DC 44 00 00 2C D2 00 00 E2 D3 01 00 14 6F 02 00 3A 3F 00 00 50 8D 00 00 B5 01 01 00 32 9B 00 00
1E0 A3 03 00 00 2C 01 00 00 01 00 00 00 2C 01 00 00 01 00 00 00 74 69 74 6C 65 2E 74 69 66 00 47 49
200 4D 50 20 32 2E 31 30 2E 32 30 00 00 32 30 32 32 3A 30 31 3A 31 37 20 30 35 3A 31 33 3A 32 34 00
220 01 00 01 00 01 00 01 00 00 00 02 A0 6C 63 6D 73 04 30 00 00 6D 6E 74 72 52 47 42 20 58 59 5A 20
240 07 E6 00 01 00 11 00 03 00 0C 00 33 61 63 73 70 4D 53 46 54 00 00 00 00 00 00 00 00 00 00 00 00
260 00 00 00 00 00 00 00 00 00 00 00 00 00 00 F6 D6 00 01 00 00 00 00 D3 2D 6C 63 6D 73 00 00 00 00
280 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
2A0 00 00 00 00 00 00 00 00 00 00 00 0D 64 65 73 63 00 00 01 20 00 00 00 40 63 70 72 74 00 00 01 60
2C0 00 00 00 36 77 74 70 74 00 00 01 98 00 00 00 14 63 68 61 64 00 00 01 AC 00 00 00 2C 72 58 59 5A
2E0 00 00 01 D8 00 00 00 14 62 58 59 5A 00 00 01 EC 00 00 00 14 67 58 59 5A 00 00 02 00 00 00 00 14
300 72 54 52 43 00 00 02 14 00 00 00 20 67 54 52 43 00 00 02 14 00 00 00 20 62 54 52 43 00 00 02 14
320 00 00 00 20 63 68 72 6D 00 00 02 34 00 00 00 24 64 6D 6E 64 00 00 02 58 00 00 00 24 64 6D 64 64
340 00 00 02 7C 00 00 00 24 6D 6C 75 63 00 00 00 00 00 00 00 01 00 00 00 0C 65 6E 55 53 00 00 00 24
360 00 00 00 1C 00 47 00 49 00 4D 00 50 00 20 00 62 00 75 00 69 00 6C 00 74 00 2D 00 69 00 6E 00 20
380 00 73 00 52 00 47 00 42 6D 6C 75 63 00 00 00 00 00 00 00 01 00 00 00 0C 65 6E 55 53 00 00 00 1A
3A0 00 00 00 1C 00 50 00 75 00 62 00 6C 00 69 00 63 00 20 00 44 00 6F 00 6D 00 61 00 69 00 6E 00 00
3C0 58 59 5A 20 00 00 00 00 00 00 F6 D6 00 01 00 00 00 00 D3 2D 73 66 33 32 00 00 00 00 00 01 0C 42
3E0 00 00 05 DE FF FF F3 25 00 00 07 93 00 00 FD 90 FF FF FB A1 FF FF FD A2 00 00 03 DC 00 00 C0 6E
```

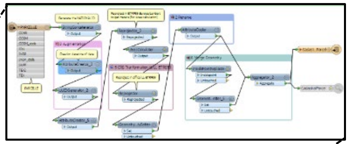


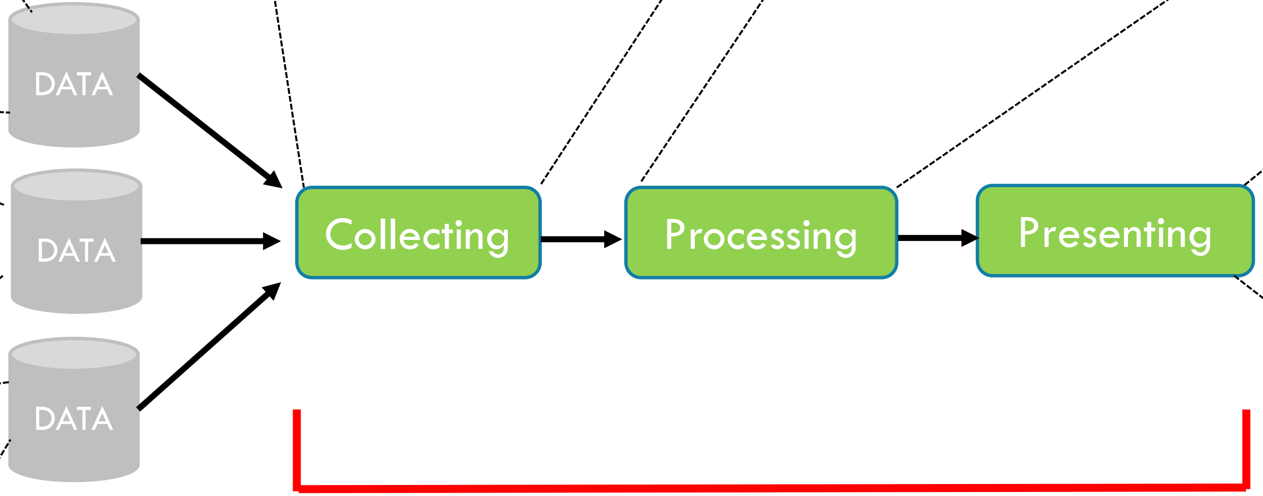
```xml
<?xml version="1.0"?>
<Example xmlns="http://www.opengis.net/gml"
         xmlns:ex="http://www.core06.mx/Ex">
<boundedBy><Box><coord><X>308787.49</X><Y>4744080.86</Y></coord>
  <coord><X>315101.36</X><Y>4748098.29</Y></coord></Box></boundedBy>
<featureMember> <ex:Municipality>
  <ex:id>1</ex:id><ex:name>Mazaricos</ex:name>
  <ex:geo><Polygon srsName="EPSG:23031"><outerBoundaryIs>
   <LinearRing><coordinates>
      309440.29,4744357.47 309038.81,4744668.04 308787.49,4745676.36
      310118.86,4746562.93 310363.24,4747445.11 311109.89,4747560.09
      312741.84,4748098.29 313455.53,4747914.19 309440.29,4744357.47
   </coordinates></LinearRing></outerBoundaryIs></Polygon></ex:geo>
</ex:Municipality> </featureMember>
<featureMember> <ex:Municipality>
  <ex:id>2</ex:id><ex:name>Oroso</ex:name>
  <ex:geo><Polygon srsName="EPSG:4230"><outerBoundaryIs>
   <LinearRing><coordinates>
      -99.0249938,56.6880477 -99.0258258,56.687203 -99.0255803,56.6863047
      -99.0248452,56.6854873 -99.0241594,56.685209 -99.023588,56.6851012
      -99.0223964,56.6851823 -99.018169,56.6861353 -99.0249938,56.6880477
   </coordinates></LinearRing></outerBoundaryIs></Polygon></ex:geo>
 </ex:Municipality> </featureMember>
</Example>
```



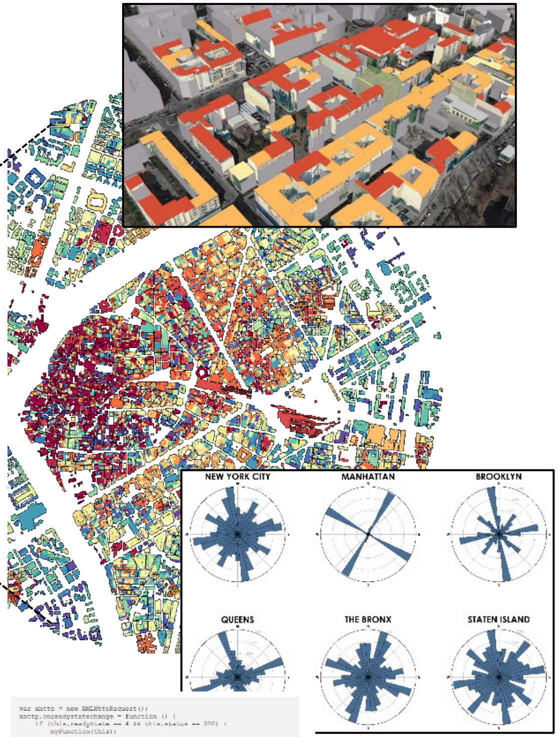
```json
{
  "Time"      : "2014-02-12 14:20:05",
  "Latitude"  : 37.33233141,
  "Longitude" : -122.0312186,
  "Count"     : 101,
  "Comments"  : "Bad Data. SNOW DAY!!",
  "Luma"      : 0,
  "Habitat"   : "Back yard, grass",
  "Types"     : [
                  "5",
                  "6"
                ],
  "Address"   : {
                  "Street"  : "2522 West Georgia Road",
                  "City"    : "Piedmont",
                  "State"   : "South Carolina",
                  "Country" : "United States",
                }
}
```







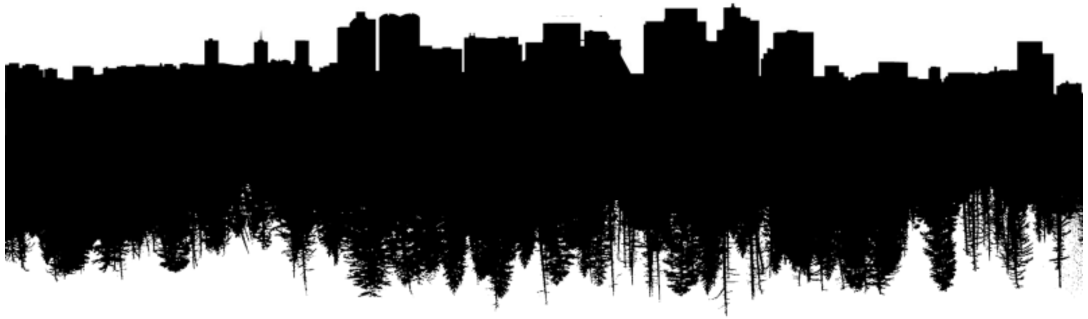
Python Notebook



```javascript
var xhttp = new XMLHttpRequest();
xhttp.onreadystatechange = function () {
    if (this.readyState == 4 && this.status == 200) {
        myFunction(this);
    }
};
xhttp.open("GET", "XMLFile1.xml", true);
xhttp.send();

function myFunction(xml) {
    var xmlDoc = xml.responseXML;
    var cityNames = Array.from(xmlDoc.getElementsByTagName("city"));
    var i = 0;

    do {
        console.log(cityNames[i].innerHTML);
        i++;
    }
    while (i < cityNames.length);
}
```



DATA BASICS CHEAT SHEET

UNDERSTANDING DIVISIONS IN DATA

open vs. propritary

packed vs. unpacked

ASCII vs. binary

numbers vs. text

HTML vs. XML

code vs. comments

XML vs. JSON

RGB vs. CMYK

Windows vs. Mac vs. Linux

NUMBER SYSTEMS

Decimal	(10-base)	[Values: 0,1,2,3,4,5,6,7,8,9]
Binary	(2-base)	[Values: 0,1]
Octal	(8-base)	[Values: 0,1,2,3,4,5,6,7]
Hexadecimal	(16-base)	[Values: 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F]

https://www.youtube.com/watch?v=aW3qCcH6Dao

https://www.youtube.com/watch?v=GPnLy6YO-0M

DATA TYPES

Numbers

Bit & Nybble	1bit & 4 bits	(max. 2 & 16)	
Byte	8-bits	(max. 256)	Byte
Word	2 bytes, 16 bits	(max. 65 536)	Small Integer (signed/unsigned)
Double word	4 bytes, 32 bits	(max. 4 294 967 296)	Integer (signed/unsigned)
Quad word	8 bytes, 64 bits	(max.18 446 744 073 709 551 616)	Floating point values

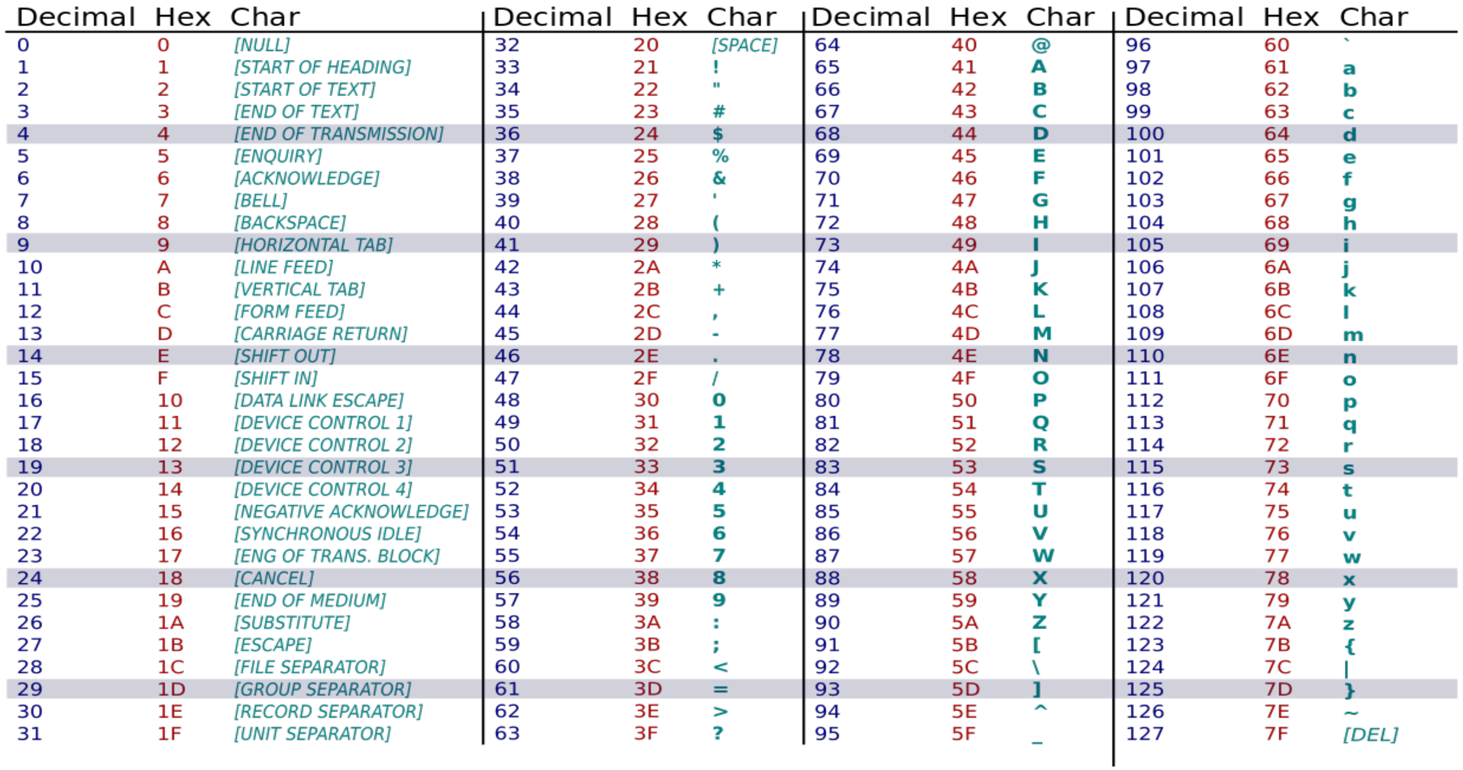
Text

ASCII/ANSI	1 byte	(max. 256)	Character
UNICODE	2 bytes	(max. 65 536)	Unicode character

Date & Time (YYYY-MM-DD hh:mm:ss)

Small datatime	4 bytes	1900-01-01 through 2079-06-06	1 minute accuracy
Datetime	8 bytes	1753-01-01 through 9999-12-31	0.00333 second accuracy

CODING TEXT



Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	`
1	1	[START OF HEADING]	33	21	!	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22	"	66	42	B	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	4	[END OF TRANSMISSION]	36	24	$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	e
6	6	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
7	7	[BELL]	39	27	'	71	47	G	103	67	g
8	8	[BACKSPACE]	40	28	(	72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29	)	73	49	I	105	69	i
10	A	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
11	B	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	44	2C	,	76	4C	L	108	6C	l
13	D	[CARRIAGE RETURN]	45	2D	-	77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	.	78	4E	N	110	6E	n
15	F	[SHIFT IN]	47	2F	/	79	4F	O	111	6F	o
16	10	[DATA LINK ESCAPE]	48	30	0	80	50	P	112	70	p
17	11	[DEVICE CONTROL 1]	49	31	1	81	51	Q	113	71	q
18	12	[DEVICE CONTROL 2]	50	32	2	82	52	R	114	72	r
19	13	[DEVICE CONTROL 3]	51	33	3	83	53	S	115	73	s
20	14	[DEVICE CONTROL 4]	52	34	4	84	54	T	116	74	t
21	15	[NEGATIVE ACKNOWLEDGE]	53	35	5	85	55	U	117	75	u
22	16	[SYNCHRONOUS IDLE]	54	36	6	86	56	V	118	76	v
23	17	[ENG OF TRANS. BLOCK]	55	37	7	87	57	W	119	77	w
24	18	[CANCEL]	56	38	8	88	58	X	120	78	x
25	19	[END OF MEDIUM]	57	39	9	89	59	Y	121	79	y
26	1A	[SUBSTITUTE]	58	3A	:	90	5A	Z	122	7A	z
27	1B	[ESCAPE]	59	3B	;	91	5B	[	123	7B	{
28	1C	[FILE SEPARATOR]	60	3C	<	92	5C	\	124	7C	|
29	1D	[GROUP SEPARATOR]	61	3D	=	93	5D	]	125	7D	}
30	1E	[RECORD SEPARATOR]	62	3E	>	94	5E	^	126	7E	~
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	_	127	7F	[DEL]

ASCII - American Standard Code for Information Interchange 7-bit

ANSI - American National Standards Institute 8-bit

Unicode (see: https://en.wikipedia.org/wiki/List_of_Unicode_characters)

Hex-to-ASCII https://www.rapidtables.com/convert/number/hex-to-ascii.html

ASCII-to-Hex https://www.rapidtables.com/convert/number/ascii-to-hex.html

CODING COLOUR

Color spaces are typically of DWORD length i.e. 4 bytes (32 bits) long.

Threfore there is 1 byte (256 values) per color component.

Additive colors (RGB):

https://www.youtube.com/watch?v=LCs8mK1rzc0

Substractive colors (CMYK):

https://www.youtube.com/watch?v=r8ejTUNwgTo

Colors in WWW: https://en.wikipedia.org/wiki/Web_colors

http://htmlcolorcodes.com/

WORLD WIDE WEB CONSORTIUM

The World Wide Web Consortium (W3C) is an international community that develops open standards to ensure the long-term growth of the Web. It is the biggest open source community.

Founded and currently led by Tim Berners-Lee, the consortium is made up of member organizations which maintain full-time staff for the purpose of working together in the development of standards for the World Wide Web. As of 24 September 2017, the World Wide Web Consortium has 463 members.

Standards:

http://www.w3.org/standards/webdesign/

http://www.w3.org/standards/xml/

HTML

Hypertext Markup Language is based on marking the logical structure of text by tagging.

For example `<em> EMPHASISED TEXT HERE </em>`

```
<!doctype html>

<html>
         <head>
                   <title> </title>
         </head>
         <body>
                   <p> Hello World </p>
         </body>

</html>

<!- COMMENTS ARE MARKED LIKE THIS -->
```

Elements in HTML: http://www.w3.org/TR/2011/WD-html5-20110525/semantics.html#semantics
HTML validator: https://validator.w3.org/#validate_by_input

CSS

Style is beyond baseline HTML structure and are defined in Cascading Style Sheets.

Can be separate files or inline coding.

In <head> section:

```
<link rel="stylesheet" type="text/css" href="mystyle.css">
```

Inside <style> tags

```
<style>

h1 {
    color: navy;
    margin-left: 20px;
}

</style>
```

Inside tags:

```
<h1 style="color:blue;margin-left:30px;">This is a heading</h1>
```

Specifications: http://www.w3.org/Style/CSS/specs.en.html

XML

```
<note>
        <to>Tove</to>
        <from>Jani</from>
        <heading>Reminder</heading>
        <body>Don't forget me this weekend!</body>
</note>
```

Extensible Markup Language (XML) is a markup language that defines a set of rules for encoding documents in a format that is both human-readable and machine-readable.

A markup language is a system for annotating a document in a way that is syntactically distinguishable from the text.

The idea and terminology evolved from the "marking up" of paper manuscripts, i.e., the revision instructions by editors, traditionally written with a blue pencil on authors' manuscripts.

Several *schema systems* exist to aid in the definition of XML-based languages.

Hundreds of document formats using XML syntax have been developed , for example GML schema for geographical data by Open Geospatial Consortium (OGC).

JSON JavaScript Object Notation

JSON is an open standard file format and data interchange format that uses human-readable text to store and transmit data objects consisting of attribute–value pairs and arrays.

JSON is and industry standard very similar to XML, but few advantages for programming.

Differences between XML and JSON include

- JSON doesn't use end tag
- JSON is shorter
- JSON is quicker to read and write
- JSON can use arrays

JSON in Wikipedia:	https://en.wikipedia.org/wiki/JSON
XML vs. JSON comparison:	https://www.w3schools.com/js/js_json_xml.asp

SVG

Scalable Vector Graphics

Inline SVG:

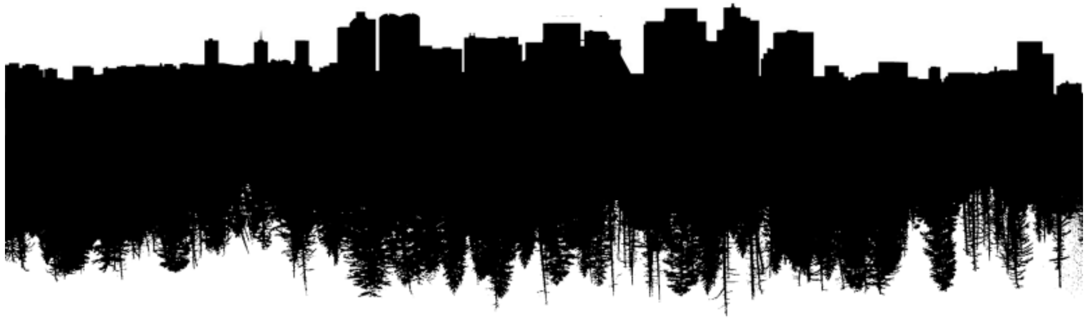
```
<div><svg><!-- WHERE THE MAGIC HAPPENS. --></svg></div>
```

EXAMPLE SYNTAX:

```
<svg height="210" width="500">
      <line x1="0" y1="0" x2="200" y2="200“

      style="stroke:rgb(255,0,0);stroke-width:2" />
</svg>
```

SVG Tutorial:	https://www.w3schools.com/graphics/svg_line.asp
SVG elements:	https://www.w3schools.com/graphics/svg_reference.asp
Official specification:	http://www.w3.org/Graphics/SVG/



ADVANCED TOPICS

DIFFERENCES IN OPERATING SYSTEMS

Coding new line i.e. pressing <ENTER>

Mac OS & Apple II family:	`0D`	(carriage return)
Linux/Unix:	`0A`	(line feed)
Windows:	`0D 0A`	(carriage return + line feed)

Memory storage for data: `90 AB 12 CD`

Little Endian (IBM): DWORD: `CD 12 AB 90` WORD `AB 90` + `CD 12`

(i.e. least significant byte to the most significant byte)

Big Endian (Sun): DWORD: `90 AB 12 CD` WORD `90 AB` + `12 CD`

(i.e. most significant byte to the least significant byte)

Tutorial: https://www.youtube.com/watch?v=T1C9Kj_78ek

TIFF

Tagged Image File Format

Image File Header

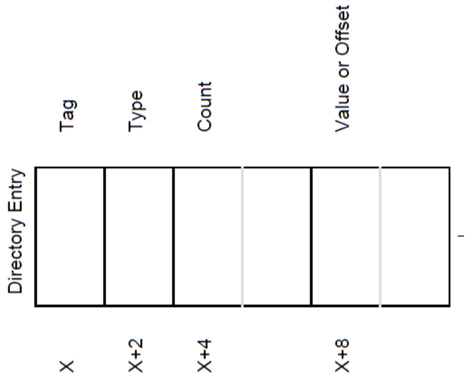
Bytes 0-1:
Byteorder
"II" (or "MM")

Bytes 2-3:
Tiff ID
"42"

Bytes 4-7:
IFD offset
"8"

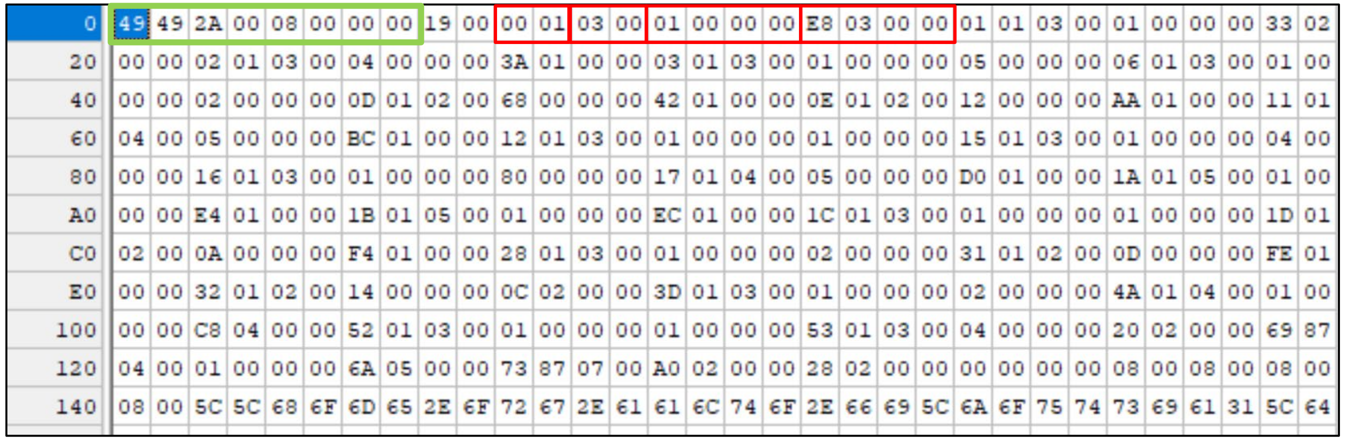
Image File Directory

Number of
Directory entries:
"25"



ImageLength
Tag = 257 (101.H)
TypeCode
3 = SHORT

SAMPLE FILE:



```
  0 49 49 2A 00 08 00 00 00 19 00 00 01 03 00 01 00 00 00 E8 03 00 00 01 01 03 00 01 00 00 00 33 02
 20 00 00 02 01 03 00 04 00 00 00 3A 01 00 00 03 01 03 00 01 00 00 00 05 00 00 00 06 01 03 00 01 00
 40 00 00 02 00 00 00 0D 01 02 00 68 00 00 00 42 01 00 00 0E 01 02 00 12 00 00 00 AA 01 00 00 11 01
 60 04 00 05 00 00 00 BC 01 00 00 12 01 03 00 01 00 00 00 01 00 00 00 15 01 03 00 01 00 00 00 04 00
 80 00 00 16 01 03 00 01 00 00 00 80 00 00 00 17 01 04 00 05 00 00 00 D0 01 00 00 1A 01 05 00 01 00
 A0 00 00 E4 01 00 00 1B 01 05 00 01 00 00 00 EC 01 00 00 1C 01 03 00 01 00 00 00 01 00 00 00 1D 01
 C0 02 00 0A 00 00 00 F4 01 00 00 28 01 03 00 01 00 00 00 02 00 00 00 31 01 02 00 0D 00 00 00 FE 01
 E0 00 00 32 01 02 00 14 00 00 00 0C 02 00 00 3D 01 03 00 01 00 00 00 02 00 00 00 4A 01 04 00 01 00
100 00 00 C8 04 00 00 52 01 03 00 01 00 00 00 01 00 00 00 53 01 03 00 04 00 00 00 20 02 00 00 69 87
120 04 00 01 00 00 00 6A 05 00 00 73 87 07 00 A0 02 00 00 28 02 00 00 00 00 00 00 08 00 08 00 08 00
140 08 00 5C 5C 68 6F 6D 65 2E 6F 72 67 2E 61 61 6C 74 6F 2E 66 69 5C 6A 6F 75 74 73 69 61 31 5C 64
```

TIFF general: https://en.wikipedia.org/wiki/TIFF

TIFF 6.0 Specification: https://www.itu.int/itudoc/itu-t/com16/tiff-fx/docs/tiff6.pdf

DATA COMPRESSION I.E. PACKING

The process of reducing the size of a data file.

Compression can be either lossy or lossless.

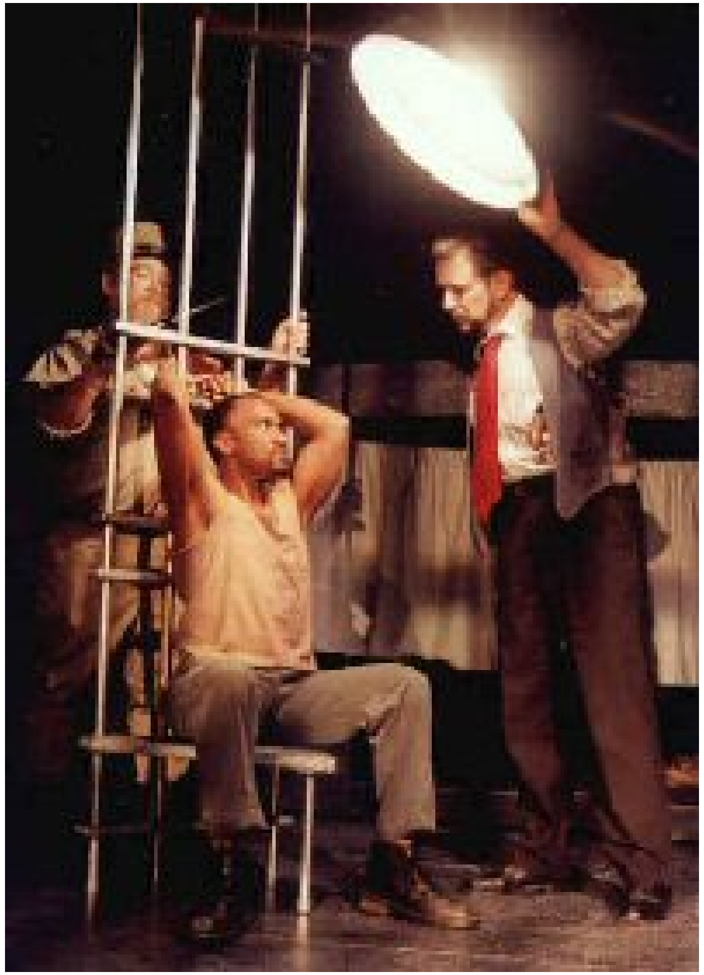
No information is lost in lossless compression. Lossy compression reduces bits by removing unnecessary or less important information.

- The Lempel–Ziv (LZ) compression methods are among the most popular algorithms for lossless storage.
- DEFLATE is a variation on LZ optimized for decompression speed and compression ratio, but compression can be slow. DEFLATE is used in PKZIP, Gzip, and PNG.
- LZW (Lempel–Ziv–Welch) is used in GIF images.
- Look for z-ending filenames: .klmz, .svgz etc.

Becoming more and more popular due to openness requirements. (*vrt. .doc vs .docx*)

MS-format specifications: https://msdn.microsoft.com/en-us/library/office/cc313105(v=office.12).aspx

QUESTIONS ?



Thank you!