cyber & data

"From bits to information"

So What Is Data?

Outline

- Data Formats.
- Unstructured, semi-structured and structured data.
- Splunk parsing.
- Bits, bytes, numbers and characters.
- Encoding and Compression.
- Magic Numbers.

"From bits to information"

Data Formats

Outline

- The Who? The What? Where? When? Why?
- Text, Documents, Images, Sounds, Video.
- Local dates: 5 Mar 2020, 5/03/2020, 03/5/20.

cyber & data

"From bits to information"

Unstructured, Semi-Structured and Structured.

Outline

- Unstructured. Within this type of data, there is no real formal structure defined for the data elements, and where we must use keywords, and to provide pointers to interesting data elements.
- Semi-structured. This type of data has some defined structure, and includes tags and/or markers that identify the semantic elements, or which provide a form of hierarchy in the data. Examples of this form include XML and JSON, and where there is some structure, but where it is not bound to a formalised data definition. The structure itself does not have to be in a table form, But could use a NoSQL (non-SQL) approach to storing data elements. Within NoSQL we can have multiple formats for our data, such as using key-value data elements.
- Structured. This defines a formal schema and has associated data models or a formal relational structure. Included within this are the encoding methods used and the fundamental definition of the entities, and the relationships between each of the entities and actions.

Unstructured and Structured

- Unstructured: Bob Smith is male and lives at 10 Cyber Avenue, and has an email address of bob@cyber. Alice McKay resides at 20 Cyber Road. She is female, and you can contact her at alice@home.

GivenName	FamilyName	Address	Email	Gender
Bob	Smith	10 Cyber Avenue	bob@home	Male
Alice	McKay	20 Cyber Road	alice@home	Female

Splunk Parsing

```
209.160.24.63 - - [11/Mar/2014:18:22:16] "GET /product.screen?productId=
    WC-SH-A02&JSESSIONID=SD0SL6FF7ADFF4953 HTTP 1.1" 200 3878 "http://
    www.google.com" "Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit
    /536.5 (KHTML, like Gecko) Chrome/19.0.1084.46 Safari/536.5" 349

209.160.24.63 - - [11/Mar/2014:18:22:16] "GET /c
    JSESSIONID=SD0SL6FF7ADFF4953 HTTP 1.1" 200
    buttercupgames.com/oldlink?itemId=EST-6" "M
    6.1; WOW64) AppleWebKit/536.5 (KHTML, like
    /19.0.1084.46 Safari/536.5" 731

209.160.24.63 - - [11/Mar/2014:18:22:17] "GET /p
    BS-AG-G09&JSESSIONID=SD0SL6FF7ADFF4953 HTTP
    www.buttercupgames.com/product.screen?produ
    /5.0 (Windows NT 6.1; WOW64) AppleWebKit/53
    Chrome/19.0.1084.46 Safari/536.5" 422
```

4/26/14
6:22:16.000 PM

```
91.205.189.15 - - [26/Apr/2014:18:22:16] "GET /oldlink?itemId=EST-14&JSESSIONID=SD6SL7FF7ADFF53113 HTTP 1.1" 200 1665 "
http://www.buttercupgames.com/oldlink?itemId=EST-14" "Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/536.5 (KHTML, lik
e Gecko) Chrome/19.0.1084.46 Safari/536.5" 159
```

Event Actions

Type	Field	Value	Actions
Event	JSESSIONID	SD6SL7FF7ADFF53113	
	bytes	1665	
	clientip	91.205.189.15	
	file	oldlink	
	ident	-	
	itemId	EST-14	
	method	GET	
	other	159	
	referer	http://www.buttercupgames.com/oldlink?itemId=EST-14	
	referer_domain	http://www.buttercupgames.com	
	req_time	26/Apr/2014:18:22:16	
	status	200	
	uri	/oldlink?itemId=EST-14&JSESSIONID=SD6SL7FF7ADFF53113	
	uri_path	/oldlink	
	uri_query	itemId=EST-14&JSESSIONID=SD6SL7FF7ADFF53113	
	user	-	
	useragent	Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/536.5 (KHTML, like Gecko) Chrome/19.0.1084.46 Safari/536.5	
	version	1.1	
Time	_time	2014-04-26T18:22:16.000+01:00	

CSV, YAML and JSON

```
Givenname,FamilyName,Address,Email,Gender
Bob,Smith,10 Cyber Avenue,bob@home,Male
Alice,McKay,"20, Cyber Road",alice@home,Female
```

```
- Givenname: Bob
  Familyname: Smith
  Address: 10 Cyber Avenue
  Email: bob@home
  Gender: Male
- Givenname: Alice
  Familyname: McKay
  Address: 20 Cyber Road
  Email: alice@home
  Gender: Female
```

```
[
    {"Givenname": "Bob", "Familyname": "Smith", "Address": "10 Cyber
        Avenue", "Email": "bob@home", "Gender": "Male"},
    {"Givenname": "Alice", "Familyname": "McKay", "Address": "20 Cyber
        Road", "Email": "alice@home", "Gender": "Female"}
]
```

```
GET /index.html
Host: 192.168.0.1
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:12.0) Gecko/20100101
    Firefox/12.0
If-Modified-Since: Sat, 03 Aug 2019 17:43:30 GMT
```

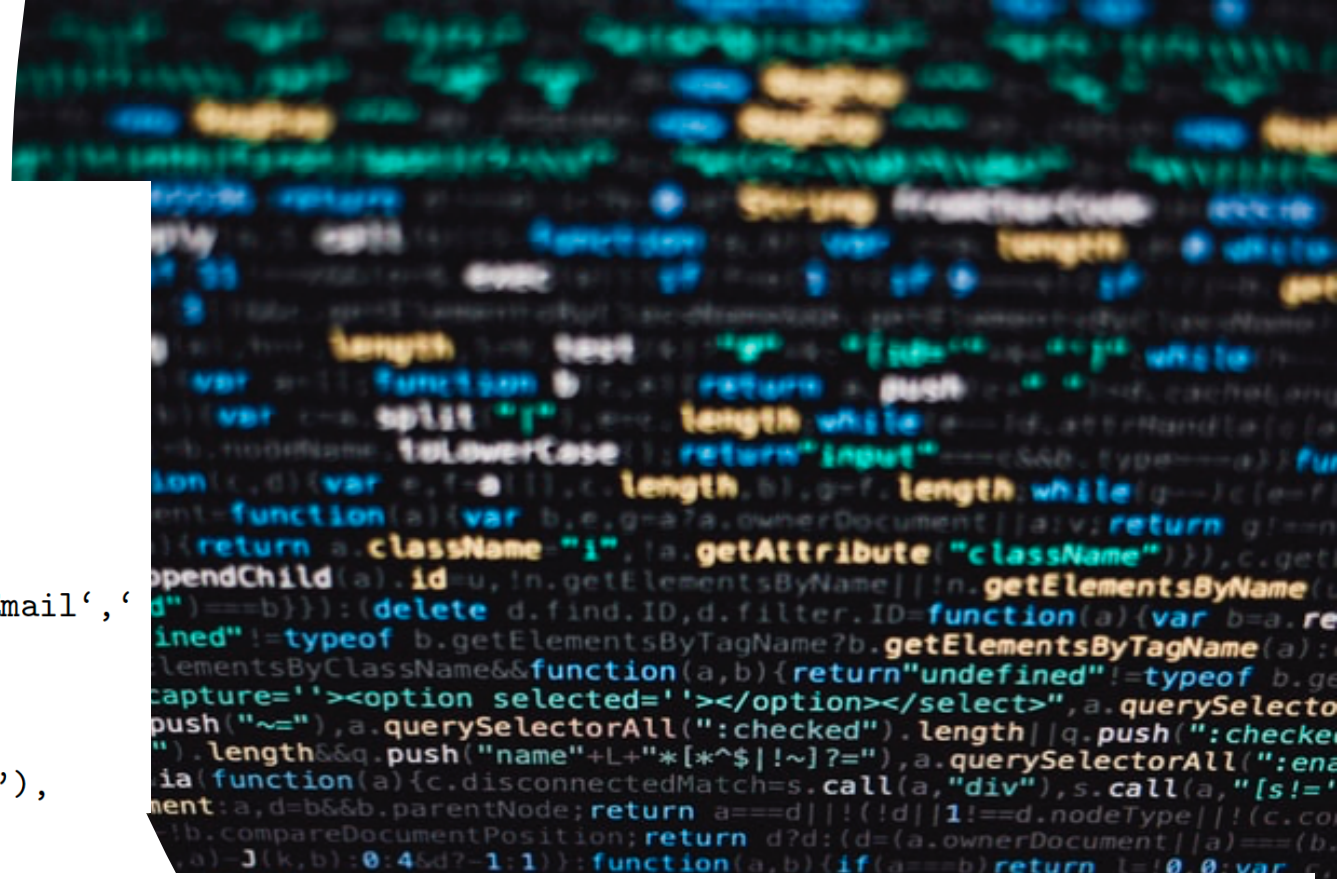
```
<p>Testing</p>
X-XSS-Protection: 0
X-Frame-Options: SAMEORIGIN
Cache-Control: private, max-age=0
Content-Type: text/html; charset=ISO-8859-1
Date: Sun, 04 Aug 2019 13:22:39 GMT
Server: gws
Accept-Ranges: none
Vary: Accept-Encoding
Transfer-Encoding: chunked
```

cyber & data

SQL and XML

```
CREATE TABLE table_name
(
    ‘Givenname‘ varchar(300),
    ‘Familyname‘ varchar(300),
    ‘Address‘ varchar(300),
    ‘Email‘ varchar(300),
    ‘Gender‘ varchar(255)
);

INSERT INTO table_name (‘Givenname‘,‘Familyname‘,‘Address‘,‘Email‘,‘
    Gender‘)
VALUES
    (’Bob’, ’Smith’, ’10 Cyber Avenue’, ’bob@home’, ’Male’),
    (’Alice’, ’McKay’, ’20 Cyber Road’, ’alice@home’, ’Female’),
```



```
<?xml version="1.0" encoding="UTF-8" ?>
<root>
    <row><Firstname>Bob</Firstname><Familyname>Smith</Familyname><Address
        >10 Cyber Avenue</Address><Email>bob@home</Email><Gender>Male</
        Gender></row>
    <row><Firstname>Alice</Firstname><Familyname>McKay</Familyname><
        Address>20 Cyber Road</Address><Email>alice@home</Email><Gender>
        Female</Gender></row>
    <row><Firstname></Firstname><Familyname></Familyname><Address></
        Address><Email></Email><Gender></Gender></row>
    <row><Firstname></Firstname><Familyname></Familyname><Address></
        Address><Email></Email><Gender></Gender></row>
</root>
```

HTML and LaTeX

```
<table>
    <tr>
        <td>Givenname</td>
        <td>Familyname</td>
        <td>Address</td>
        <td>Email</td>
        <td>Gender</td>
    </tr>
    <tr>
        <td>Bob</td>
        <td>Smith</td>
        <td>10 Cyber Avenue</td>
        <td>bob@home</td>
        <td>Male</td>
    </tr>
    <tr>
        <td>Alice</td>
        <td>McKay</td>
        <td>20 Cyber Road</td>
        <td>alice@home</td>
        <td>Female</td>
    </tr>

</table>
```

```
\begin{table}
\centering
\caption{TableName}
\begin{tabular}{|l|l|l|l|}
\hline

Givenname & Familyname & Address & Email & Gender \\ \hline
Bob & Smith & 10 Cyber Avenue & bob@home & Male \\ \hline
Alice & McKay & 20 Cyber Road & alice@home & Female \\ \hline
 &  &  &  \\ \hline
 &  &  &  \\ \hline
 \end{tabular}
```

```
firstname: Bob. Type: Text.
familyname: Smith. Type: Text
address: 10 Cyber Avenue. Type: PostalAddress.
Gender: Male. Type: GenderType.
Email: bob@cyber. Type: Text
```

Schema

```
<div itemscope itemtype="http://schema.org/Person">
  <span itemprop="givenname">Bob</span>
  <span itemprop="familyname">Smith</span>
  <div itemprop="address" itemscope itemtype="http://schema.org/
      PostalAddress"
    <span itemprop="addressLocality">10 Cyber Avenue</span>,
    <span itemprop="postalCode">XYZ</span>
  </div>
   <span itemprop="email">bob@home</span>
</div>
```

"From bits to information"

Bits, Bytes, Numbers and Characters

Numbers

- Integers can be positive or negative numbers and have no fractional part. They are represented with the $\mathbb{Z}$ symbol ... -2, -1, 0, +1, +2, ...
- Rational numbers are fractions ($\mathbb{Q}$).
- Real numbers ($\mathbb{R}$) include both integers and rational numbers, and any other number that can be used in a comparison.
- Prime numbers ($\mathbb{P}$) represent the integers which can only be divisible by itself and unity.
- Natural numbers ($\mathbb{N}$) represent positive numbers which are integers 1,2,....

Numbers

- char (byte). This uses eights bits and ranges from 0 to 255.
- signed char (char). This uses eights bits and ranges from -127 to 128.
- short (short). This uses 16 bits and ranges from -32,768 to 32,767.
- unsigned short (ushort). This uses 16 bits and ranges from 0 to 65,535.
- int (int). This uses 32 bits and ranges from -2,147,483,648 to 2,147,483,647.
- unsigned int (uint). This uses 32 bits and ranges from 0 to 4,294,967,295.
- long (long). This uses 64 bits and ranges from -9,223,372,036,854,775,808 to 9,223,372,036,854,775,807.
- unsigned long (ulong). This uses 64 bits and ranges from 0 to 18,446,744,073, 709,551,615.

Little and Big Endian

```
Location (100h): 01 (Most significant byte)
Location (101h): 02
Location (102h): 03
Location (103h): 04 (Least significant byte - at the end)
```

"From bits to information"

Encoding and Compression

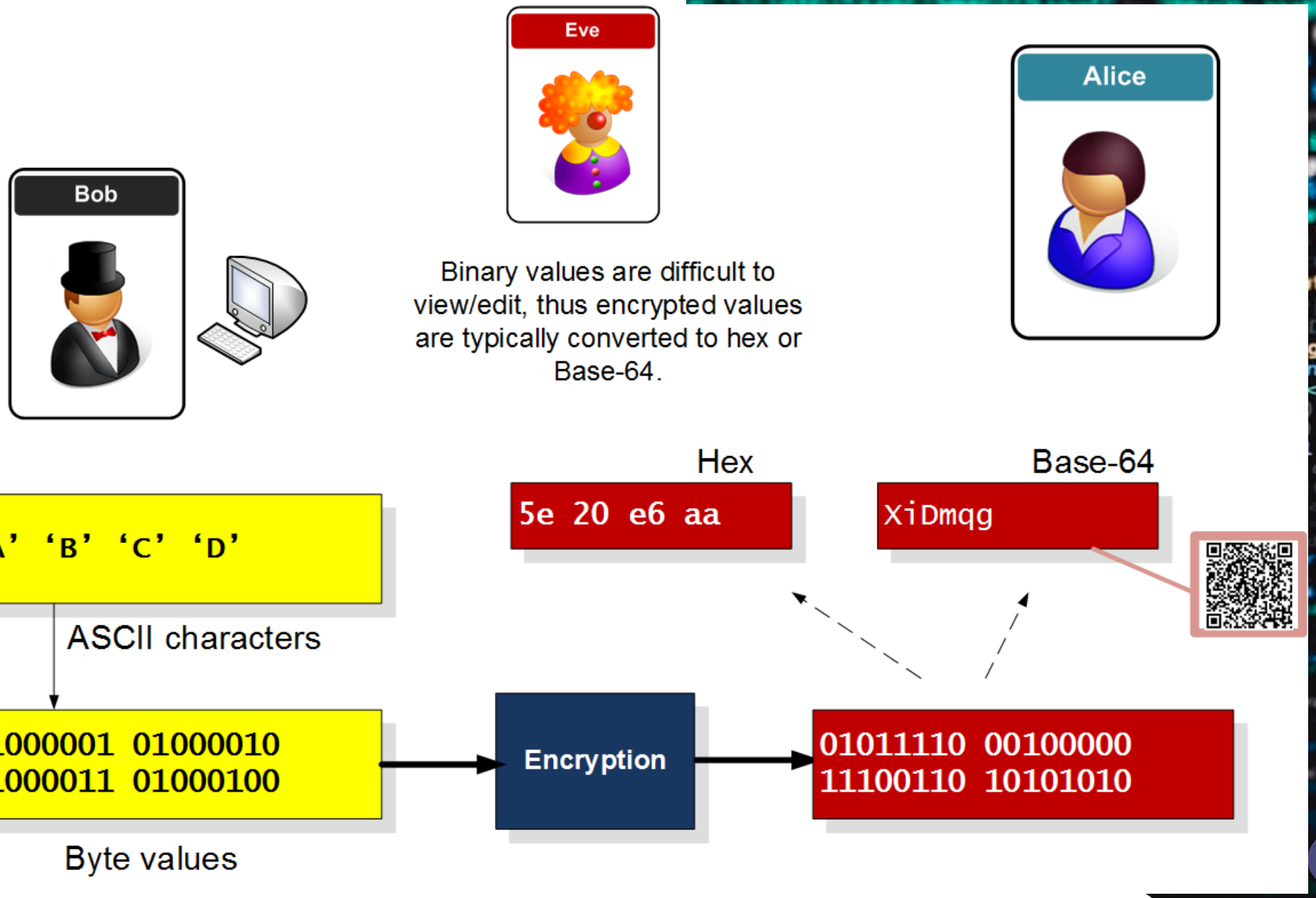
ASCII, binary, hex, …

ASCII	Binary	Hex	Decimal
é´	0110 0101	0x65	101
É´	0100 0101	0x45	69
’ ’	0010 0000	0x20	32

ASCII	Binary	Hex	Decimal	Character representation
CR	0110 0101	0x0D	13	\r
LF	0100 0101	0x0A	10	\n
HT	0000 0111	0x07	7	\t

Char	Dec	UTC-16	ASCII	Hex	Oct	HTML
A	65	00000000 01000001	01000001	41	101	&65;
B	66	00000000 01000010	01000010	42	102	&66;

ASCII, binary, hex, …



Hex and Base-64

ASCII	f	r	e	d
Binary	01100110	01110010	01100101	01100100

Binary	011001	100111	001001	100101	011001	00

Binary	011001	100111	001001	100101	011001	00
Decimal	25	39	9	37	25	0
Base-64	Z	n	J	l	Z	A

Hex

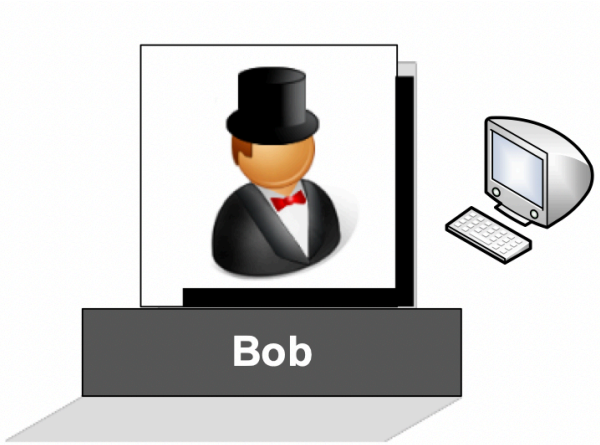
With hexadecimal, the bit stream is split into groups of four, and converted into hex values (0-9,A-F)

Bit stream

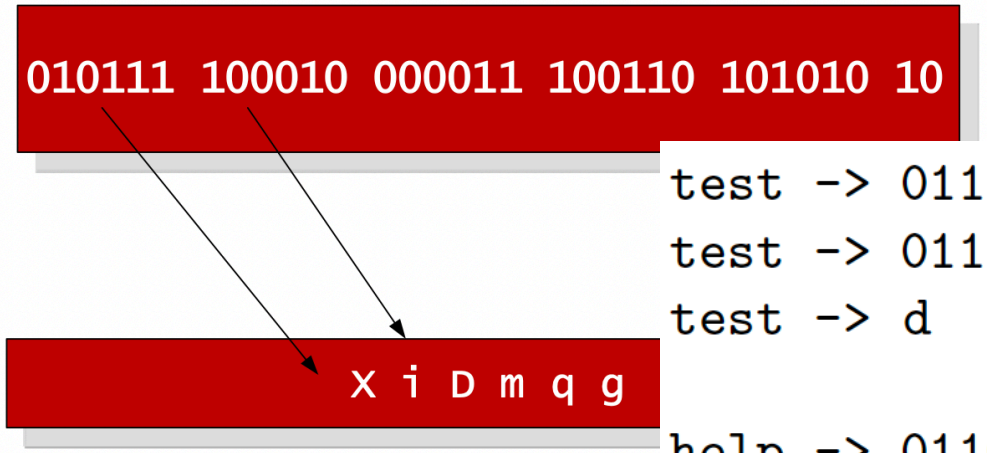
Hex

Decimal	Binary	Hex
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

Base-64



Val	Enc	Val	Enc	Val	Enc	Val	Enc
0	A	16	Q	32	g	48	w
1	B	17	R	33	h	49	x
2	C	18	S	34	i	50	y
3	D	19	T	35	j	51	z
4	E	20	U	36	k	52	0
5	F	21	V	37	l	53	1
6	G	22	W	38	m	54	2
7	H	23	X	39	n	55	3




```
test -> 01110100 01100101 01110011 01110100
test -> 011101 000110 010101 110011 011101 00[0000] = =
test ->  d      G      V      z      d   A        = =

help -> 01101000 01100101 01101100 01110000
help -> 011101 000110 010101 110011 011101 00[0000] = =
help ->  a     G      V      s      c   A        = =
```

Compression

```
Input: hello
Compressed: eJzLSM3JyQcABiwCFQ==
Compressed: <Buffer 78 9c cb 48 cd c9 c9 07 00 06 2c 02 15>
```

```
Input: eJwLSS0uMTQyBgAJ
Uncompressed: Test123
Uncompressed: <Buffer 5
```

```
Input: abcabcabcabcabcab
abcabcabcabcabcabcabcabca
Compressed: eJxLTEp0pC8CA
```

```
Input: Go learn some cryp
Compressed:
    eJxzz1fISU0sylMozs9NV
```

```
var zlib = require('zlib');
var test="hello";
 var input = new Buffer.from(test)
zlib.deflate(input, function(err, buf) {
  var res=buf.toString('base64');
console.log("Compressed: " ,res );
 // console.log("Compressed: " ,buf );
});
}
else {
 var input = new Buffer.from(test,'base64')

 zlib.inflate(input, function(err, buf) {
      console.log("Uncompressed:", buf.toString("utf8") );
 // console.log("Uncompressed: " ,buf );
});
}
```

"From bits to information"

Magic Numbers

Magic Numbers

Description	Extension	Magic Number
Adobe Illustrator	.ai	25 50 44 46 [%PDF]
Bitmap graphic	.bmp	42 4D [BM]
Class File	.class	CA FE BA BE
JPEG graphic file	.jpg	FFD8
JPEG 2000 graphic file	.jp2	0000000C6A5020200D0A [....jP..]
GIF graphic file	.gif	47 49 46 38 [GIF89]
TIF graphic file	.tif	49 49 [II]
PNG graphic file	.png	89 50 4E 47 .PNG
WAV audio file	.png	52 49 46 46 RIFF
ELF Linux EXE	.png	7F 45 4C 46 .ELF
Photoshop Graphics	.psd	38 42 50 53 [8BPS]
Windows Meta File	.wmf	D7 CD C6 9A
MIDI file	.mid	4D 54 68 64 [MThd]
Icon file	.ico	00 00 01 00
MP3 file with ID3 identity tag	.mp3	49 44 33 [ID3]
AVI video file	.avi	52 49 46 46 [RIFF]
Flash Shockwave	.swf	46 57 53 [FWS]
Flash Video	.flv	46 4C 56 [FLV]
Mpeg 4 video file	.mp4	00 00 00 18 66 74 79 70 6D 70 34 32 [....ftypmp42]
MOV video file	.mov	6D 6F 6F 76 [....moov]
Windows Video file	.wmv	30 26 B2 75 8E 66 CF

Gzip, PNG and GIF

```
[00000000] 1F 8B 08 08 B5 7B B6 50 00 0B 74 65 73 74 2E 74   .....{.P..test.t
[00000016] 78 74 00 0B C9 C8 2C 56 00 A2 44 85 92 D4 E2 12   xt....,V..D.....
[00000032] 3D 20 00 00 33 F4 72 66 12 00 00 00
```

```
[00000000] 89 50 4E 47 0D 0A 1A 0A .PNG....
[00000008] 00 00 00 0D 49 48 44 52 ....IHDR
[00000016] 00 00 00 F3 00 00 00 C3 ........
[00000024] 08 06 00 00 00 57 8C 27 .....W.'
[00000032] 92 00 00 00 04 67 41 4D .....gAM
[00000040] 41 00 00 AF C8 37 05 8A A....7..
[00000048] E9 00 00 00 19 74 45 58 .....tEX
```

```
[00000000] 47 49 46 38 39 61 64 00 GIF89ad.
[00000008] 55 00 E6 00 00 FF FF FF U.......
[00000016] F7 F7 F6 F1 F4 F2 EE EE ........
[00000024] EF E7 E7 E7 E1 E4 E6 DF ........
```

PKZip and Office XML

```
[00000000] 50 4B 03 04 14 00 02 00 PK......
[00000008] 08 00 80 9D 6C 39 DA 4D ....l9.M
```

```
[00000000] 50 4B 03 04 14 00 06 00 PK......
[00000008] 08 00 00 00 21 00 09 24 ....!..$
[00000016] 87 82 81 01 00 00 8E 05 ........
[00000024] 00 00 13 00 08 02 5B 43 ......[C
[00000032] 6F 6E 74 65 6E 74 5F 54 ontent_T
[00000040] 79 70 65 73 5D 2E 78 6D ypes].xm
[00000048] 6C 20 A2 04 02 28 A0 00 l....(..
```

```
Version: 14 00
General purpose bit flag: 02 00
Compression method: 08 00
File last modification time: 80 9D
File last modification date: 6C 39
CRC: DA4DB80F
Compessed size: 90010000
Uncompressed size: 27060000
File name length: 0900
Extra field length: 0000
Filename: anim.xaml
```

cyber & data

"From bits to information"

So What Is Data?