CS102: Data Types

Hannah Aizenman

haizenm00@ccny.cuny.edu

Information is packaged in all sorts of ways...

Text

*...it became instantly clear why inundation was a serious problem ... more than 80% of the data is missing as measured by 100*isNAN/datapoints...*

Numbers

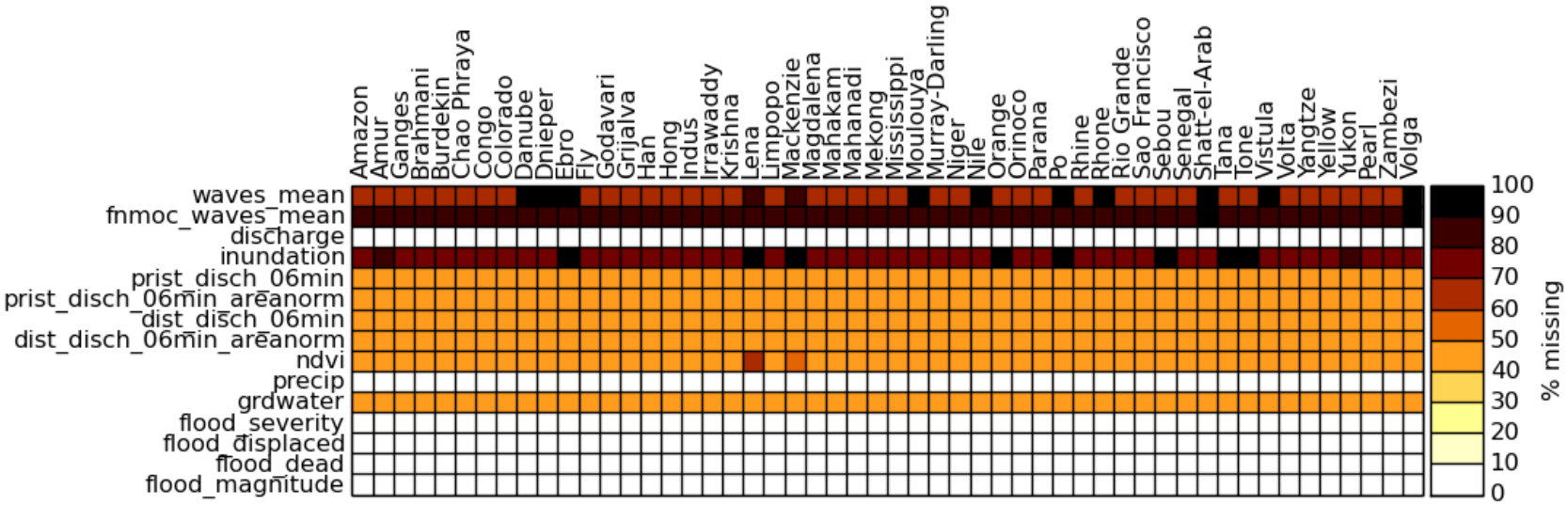
missing.csv - OpenOffice.org Calc

File Edit View Insert Format Tools Data Window Help

	A	B	C	D	E	F	
1		waves_mean	fnmoc_waves_mean	discharge	inundation	prist_disch_06min	prist_disch_
2	Amazon	%69.24	%82.54	%8.70	%70.63	%43.48	%43.48
3	Amur	%69.27	%82.54	%8.70	%88.01	%43.48	%43.48
4	Ganges	%69.24	%82.54	%8.70	%70.63	%43.48	%43.48
5	Brahmani	%69.24	%82.54	%8.70	%70.63	%43.48	%43.48
6	Burdekin	%69.24	%82.54	%8.70	%70.63	%43.48	%43.48
7	Chao Phraya	%69.24	%82.54	%8.70	%70.63	%43.48	%43.48
8	Congo	%69.24	%82.54	%8.70	%70.63	%43.48	%43.48
9	Colorado	%69.24	%82.54	%8.70	%70.63	%43.48	%43.48
10	Danube	%100.00	%88.04	%8.70	%74.16	%43.48	%43.48
11	Dnieper	%100.00	%88.04	%8.70	%76.16	%43.48	%43.48
12	Ebro	%100.00	%83.42	%8.70	%100.00	%43.48	%43.48
13	Fly	%69.24	%82.54	%8.70	%70.86	%43.48	%43.48
14	Godavari	%69.24	%82.54	%8.70	%70.63	%43.48	%43.48
15	Grijalva	%69.24	%82.54	%8.70	%70.63	%43.48	%43.48
16	Han	%69.24	%82.54	%8.70	%77.60	%43.48	%43.48
17	Hong	%69.24	%82.54	%8.70	%70.63	%43.48	%43.48
18	Indus	%69.24	%82.54	%8.70	%70.63	%43.48	%43.48

Sheet1

Pictures


Amazon
Amur
Ganges
Brahmani
Burdekin
Chao Phraya
Congo
Colorado
Danube
Dnieper
Ebro
Fly
Godavari
Grijalva
Han
Hong
Indus
Irrawaddy
Krishna
Lena
Limpopo
Mackenzie
Magdalena
Mahakam
Mahanadi
Mekong
Mississippi
Mouloiya
Murray-Darling
Niger
Nile
Orange
Orinoco
Parana
Po
Rhine
Rhone
Rio Grande
Sao Francisco
Sebou
Senegal
Shatt-el-Arab
Tana
Tone
Vistula
Volta
Yangtze
Yellow
Yukon
Pearl
Zambezi
Volga
waves_mean
fnmoc_waves_mean
discharge
inundation
prist_disch_06min
prist_disch_06min_areanorm
dist_disch_06min
dist_disch_06min_areanorm
ndvi
precip
grdwater
flood_severity
flood_displaced
flood_dead
flood_magnitude
100
90
80
70
60
50
40
30
20
10
0
% missing

...so how do computers represent information?

Binary

- computers store information using digital components

Binary

- computers store information using digital components
- these components only have on/off states

Binary

- computers store information using digital components
- these components only have on/off states
- each component can store one value

Binary

- computers store information using digital components
- these components only have on/off states
- each component can store one value
- these values are called bits

Binary

- computers store information using digital components
- these components only have on/off states
- each component can store one value
- these values are called bits
- early computers stored the data using switches

Binary 0

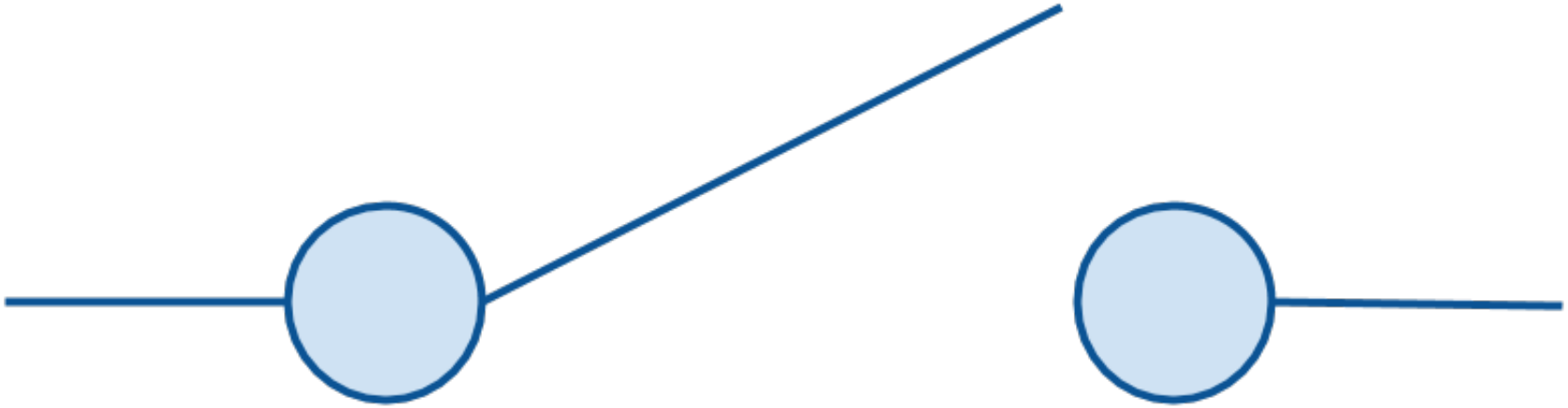


Figure : An open switch or low voltage is a binary 0

Binary 1

Figure : A closed switch or high voltage is a binary 1

Counting in Binary

Binary	
0	start
1	next
???	there's no 2!

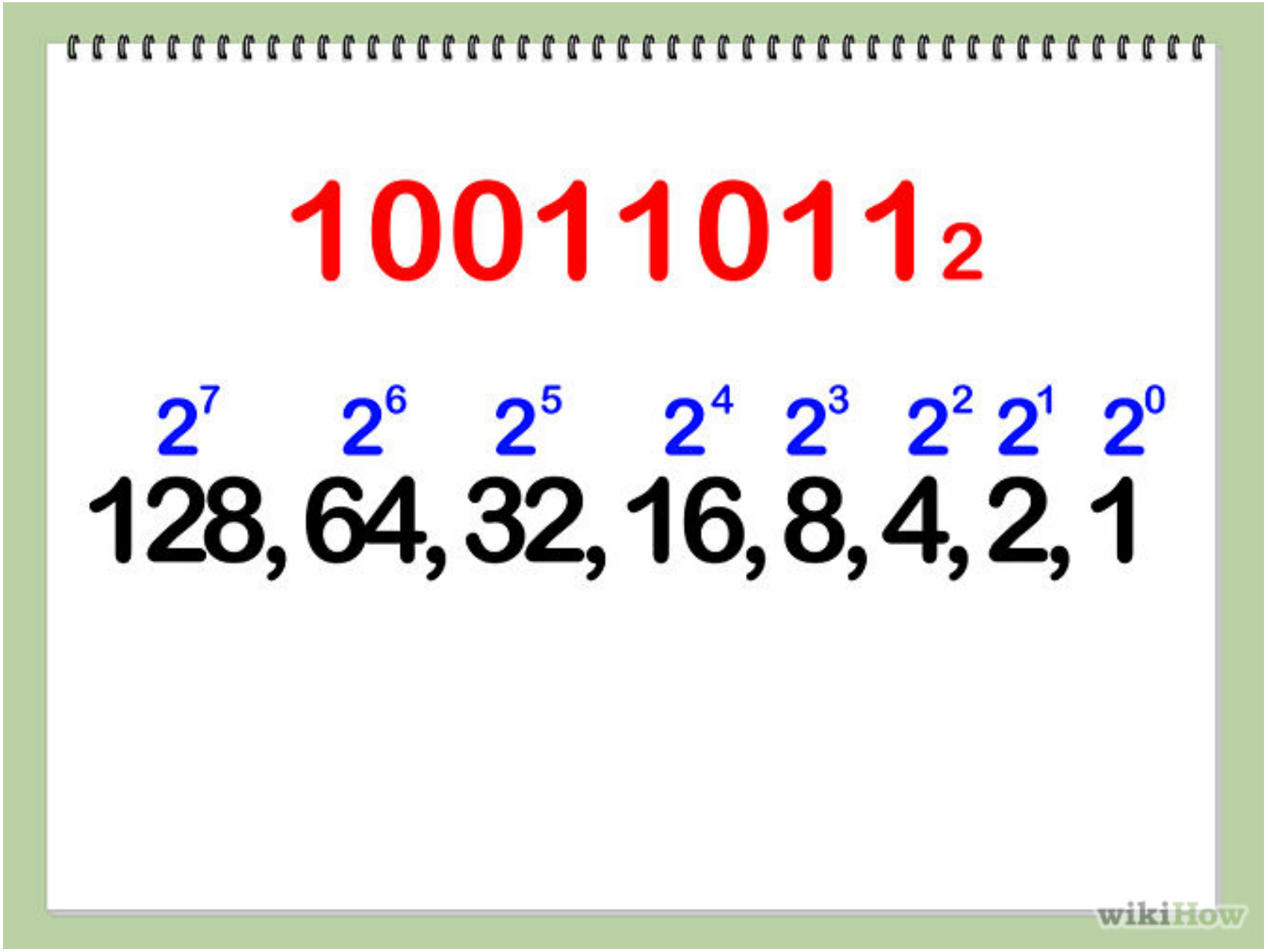
Counting in Decimal

Binary	
0	start
1, 2, ...7, 8	
9	last decimal digit
10	go back to 0, but put a 1 in the 10s place

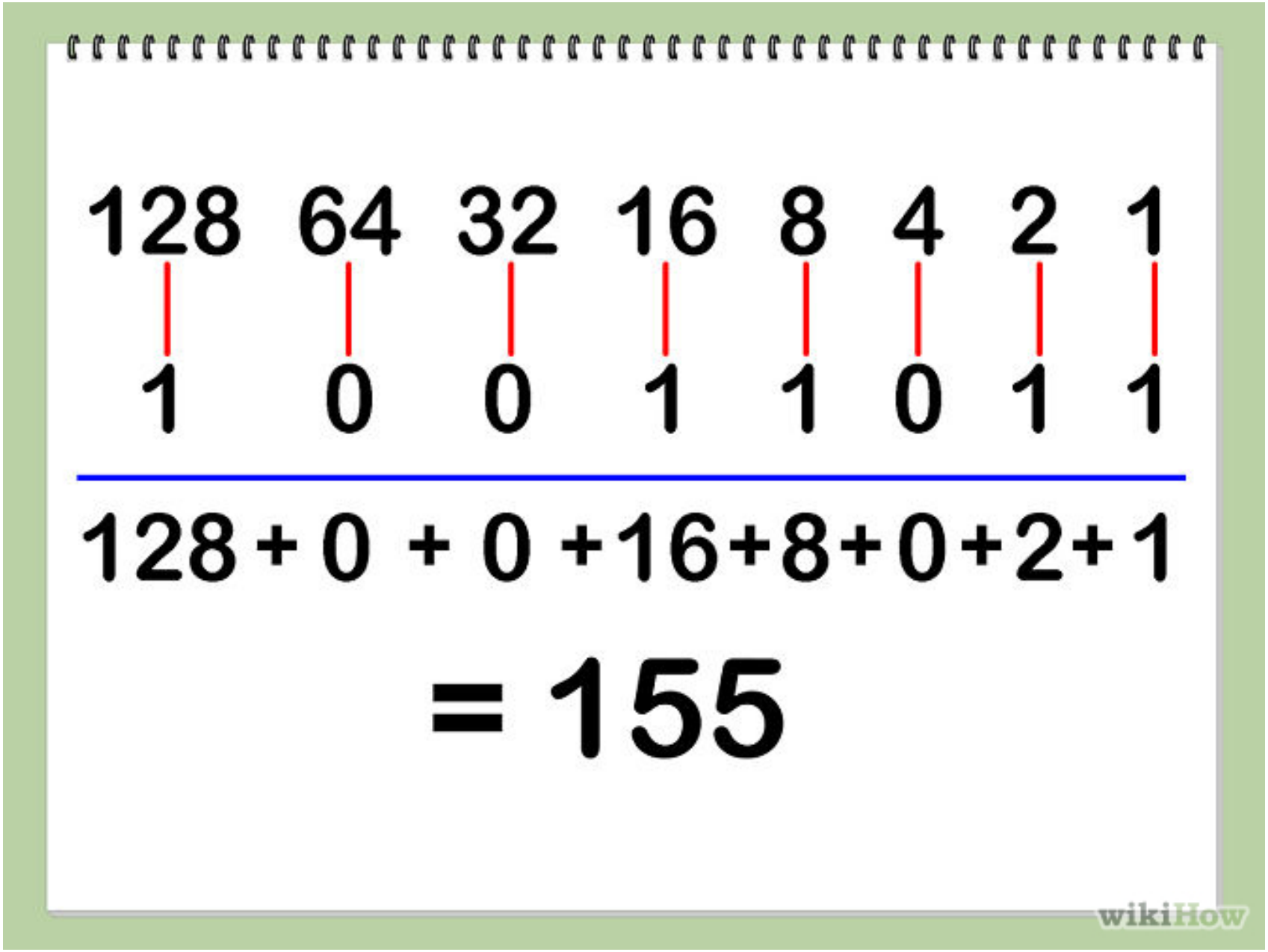
Counting in Binary

Binary	
0	start
1	next
10	back at 0, but add a 1 in the next place
11	...

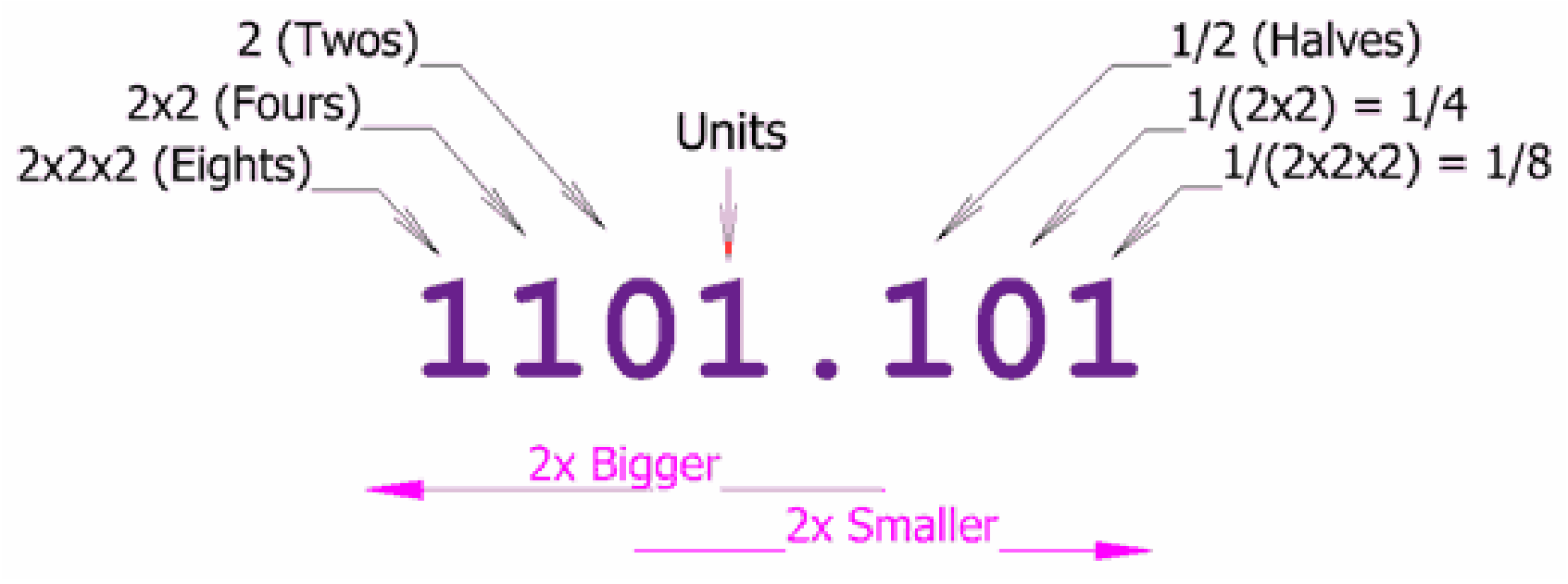
Binary: Whole Number



Binary: Whole Number



Binary: Decimal Number



Binary: Floating Point

$$\boxed{-2.3445} \times \boxed{10^3}$$

Mantissa Exponent

An 8 bit floating point number

1.001	0010
mantissa	exponent

Binary: Letter or Symbol (ASCII)

Decimal	Hexadecimal	Binary	Octal	Char	Decimal	Hexadecimal	Binary	Octal	Char
48	30	110000	60	0	96	60	1100000	140	`
49	31	110001	61	1	97	61	1100001	141	a
50	32	110010	62	2	98	62	1100010	142	b
51	33	110011	63	3	99	63	1100011	143	c
52	34	110100	64	4	100	64	1100100	144	d
53	35	110101	65	5	101	65	1100101	145	e
54	36	110110	66	6	102	66	1100110	146	f
55	37	110111	67	7	103	67	1100111	147	g
56	38	111000	70	8	104	68	1101000	150	h
57	39	111001	71	9	105	69	1101001	151	i
58	3A	111010	72	:	106	6A	1101010	152	j
59	3B	111011	73	;	107	6B	1101011	153	k
60	3C	111100	74	<	108	6C	1101100	154	l
61	3D	111101	75	=	109	6D	1101101	155	m
62	3E	111110	76	>	110	6E	1101110	156	n
63	3F	111111	77	?	111	6F	1101111	157	o
64	40	1000000	100	@	112	70	1110000	160	p
65	41	1000001	101	A	113	71	1110001	161	q
66	42	1000010	102	B	114	72	1110010	162	r
67	43	1000011	103	C	115	73	1110011	163	s
68	44	1000100	104	D	116	74	1110100	164	t
69	45	1000101	105	E	117	75	1110101	165	u
70	46	1000110	106	F	118	76	1110110	166	v
71	47	1000111	107	G	119	77	1110111	167	w
72	48	1001000	110	H	120	78	1111000	170	x
73	49	1001001	111	I	121	79	1111001	171	y
74	4A	1001010	112	J	122	7A	1111010	172	z

Datatypes

- compilers can’t tell symbols and the different types of numbers apart

Datatypes

- compilers can't tell symbols and the different types of numbers apart
- **datatypes** tell the compiler what it's working with

Datatypes

- compilers can't tell symbols and the different types of numbers apart
- **datatypes** tell the compiler what it's working with
- built-in datatypes are called **primitives**:

Datatypes

- compilers can't tell symbols and the different types of numbers apart
- **datatypes** tell the compiler what it's working with
- built-in datatypes are called **primitives**:

integer	whole number
double	decimal number
float	floating point number
character	letter or symbol
bool	true, false

Variables

Variables are used to store information for later use.

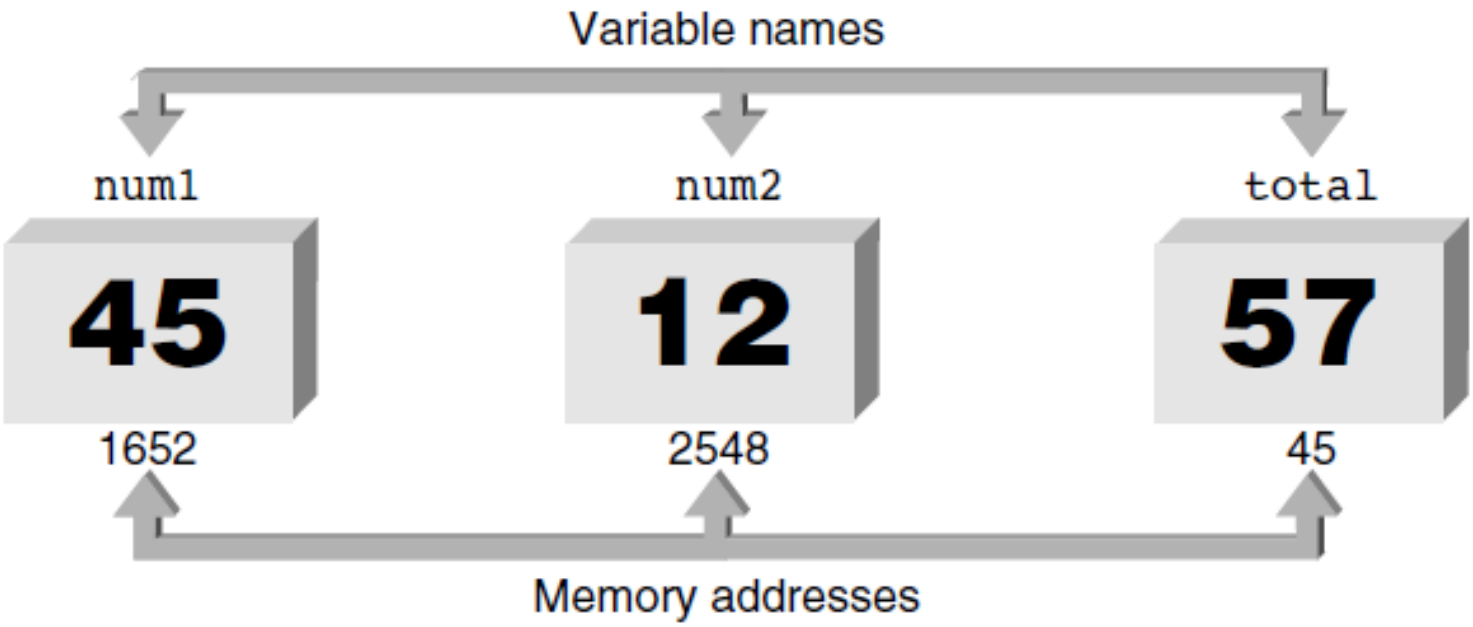


Figure : Variables consist of a name, type, and value

Variables

name how the variable is referred to throughout the code

Variables

name	how the variable is referred to throughout the code
type	the datatype of the information the variable stores

Variables

name	how the variable is referred to throughout the code
type	the datatype of the information the variable stores
value	the information the variable stores

Variable name

There are a few rules for variable names(identified)

Variable name

There are a few rules for variable names(identified)

- First character has to be a letter or underscore

good	hello, _hello
bad	1hello, @hello

Variable name

There are a few rules for variable names(identified)

- First character has to be a letter or underscore

  good hello, _hello
  bad 1hello, @hello

- Must be composed of letters, digits, and underscores

  good hello1, he1llo, he_llo
  bad h*ello, he llo

Variable name

There are a few rules for variable names(identified)

- First character has to be a letter or underscore

  good hello, _hello
  bad 1hello, @hello

- Must be composed of letters, digits, and underscores

  good hello1, he1llo, he_llo
  bad h*ello, he llo

- Can't be part of the language

  good hello, goodbye, world
  bad int, using, return

Variable name

There are a few rules for variable names(identified)

- First character has to be a letter or underscore

  good hello, _hello
  bad 1hello, @hello

- Must be composed of letters, digits, and underscores

  good hello1, he1llo, he_llo
  bad h*ello, he llo

- Can't be part of the language

  good hello, goodbye, world
  bad int, using, return

- Can't have more than 1024 characters

Declaration Statement

How they work:

- identifies the variable **name**
- specifies the variable **type**

Declaration Statement

How they work:

- identifies the variable **name**
- specifies the variable **type**

How they're written:

type *name*;

Declaration Statement

How they work:

- identifies the variable **name**
- specifies the variable **type**

How they're written:

type *name*;
type *name1*, *name2*, *name3*, ...;

Declaration: integer

- **int** *voltage*;
- **int** *power*, *resistance*;

Declaration: double

- **double** *temperature*;
- **double** *pressure*, *wind*;

Declaration: float

- **float** *acceleration*;
- **float** *gravity*, *velocity*;

Declaration: character

- **char** *no_quit*;
- **char** *no_quit*, *quit*;