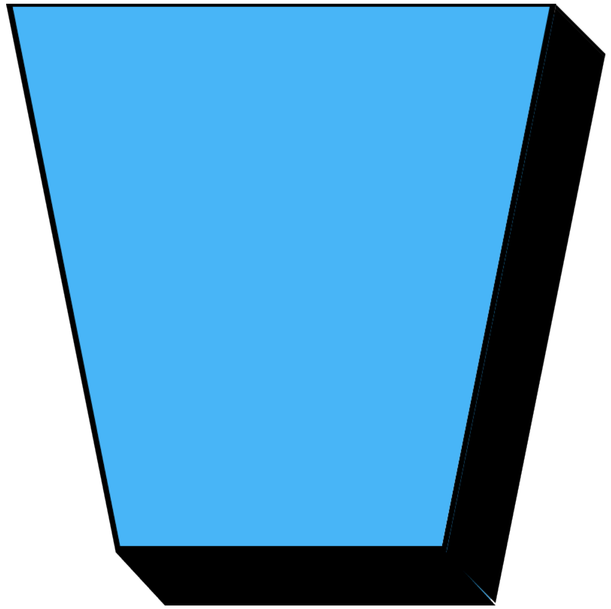
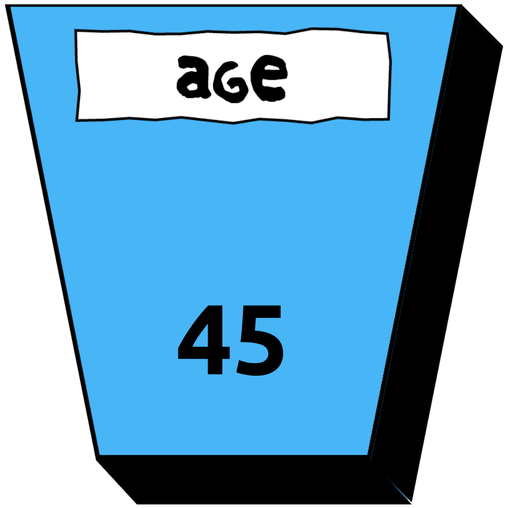
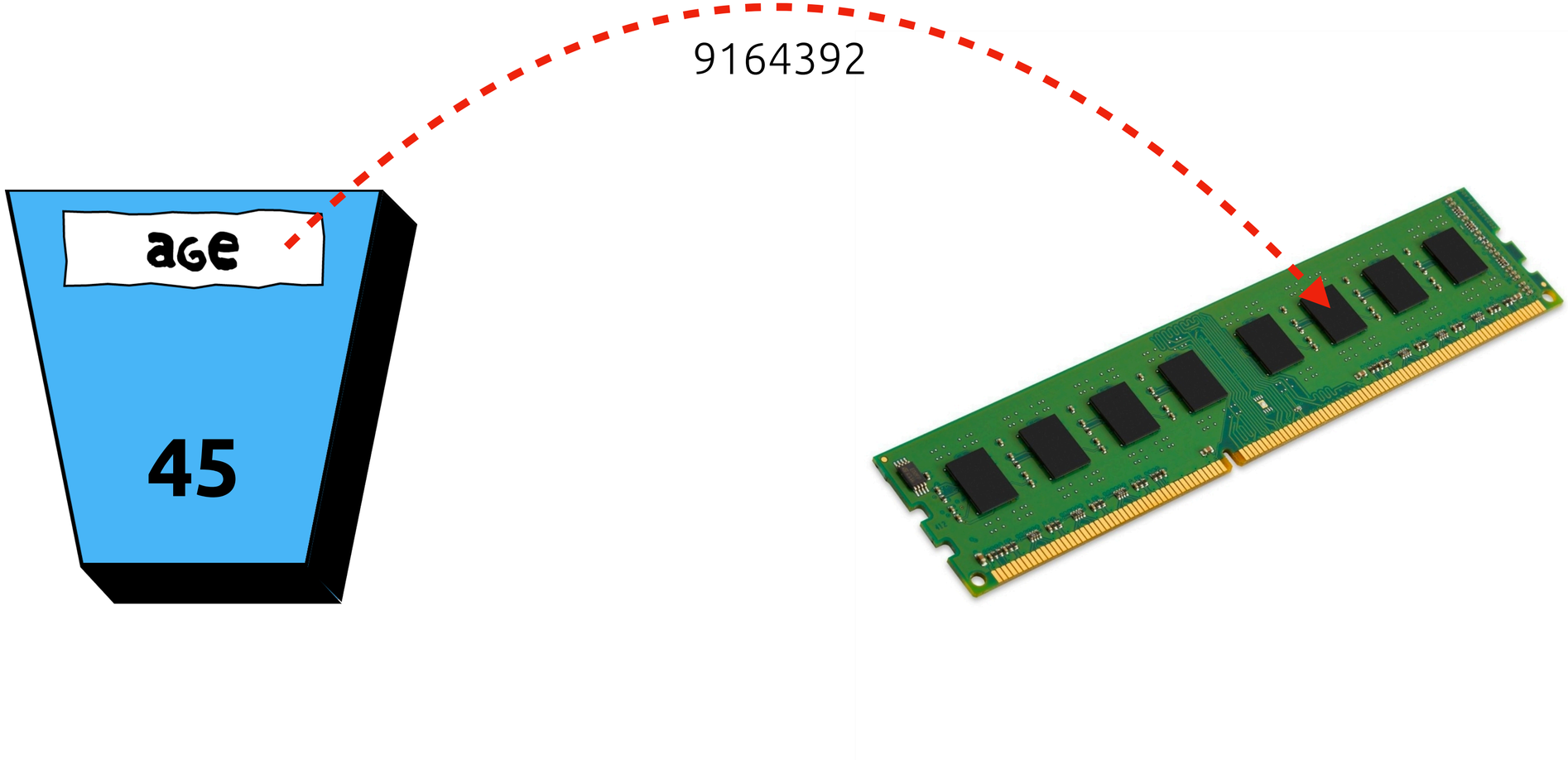
Essential Computing 1

Variables & Datatypes





a variable is like a named bin where
you can store information

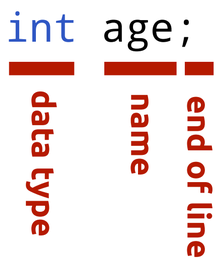


it holds an address to a location in your computer memory where the information is stored

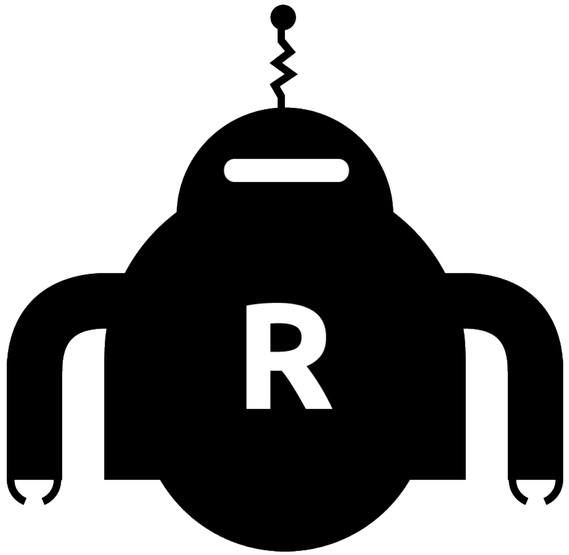
1. **Declare**
2. **Init** (assign)
3. **Use**

```
int age;
```

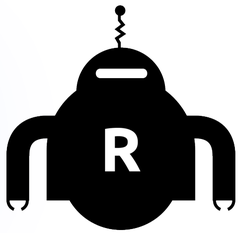
declaring a variable



declaring a variable

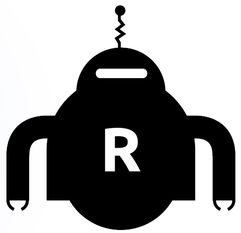


rob the robot will be reading your code



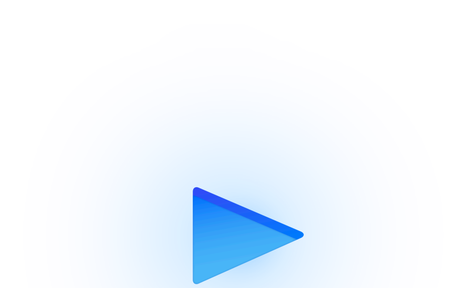
```
int age;
```

declaring a variable

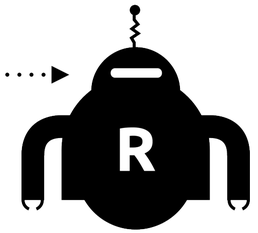


```
int age;
```

declaring a variable

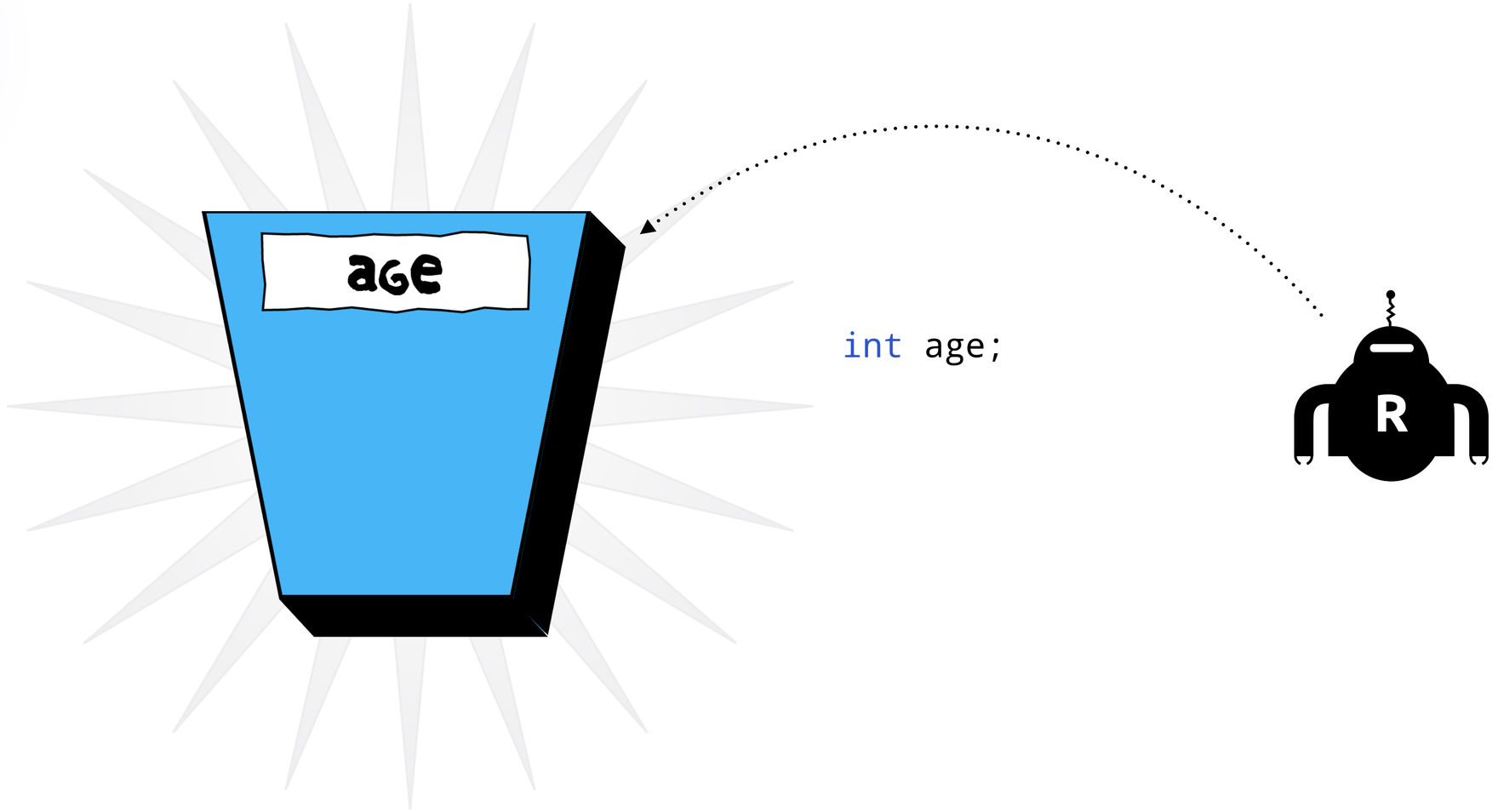


```
int age;
```




```
ok; i'll dedicate
space in the
computers memory
for a variable
named 'age'
```

declaring a variable

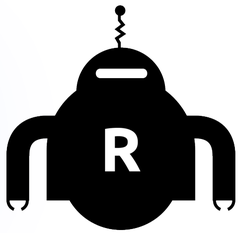


declaring a variable

```
int age;
age = 29;
```

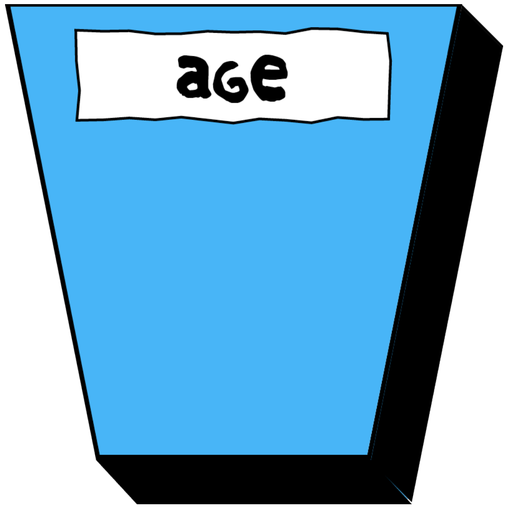
assign a value to a variable

assign a value to a variable

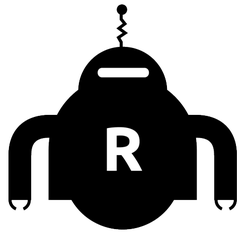



```int age;
age = 29;```

assign a value to a variable
```

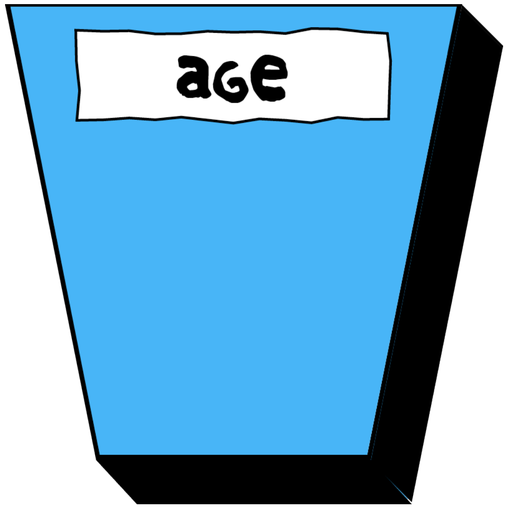



```
int age;
age = 29;
```

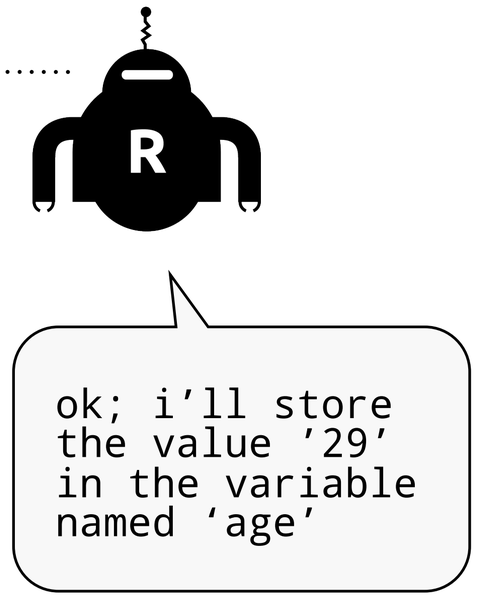


so far i've created an integer variable named age.

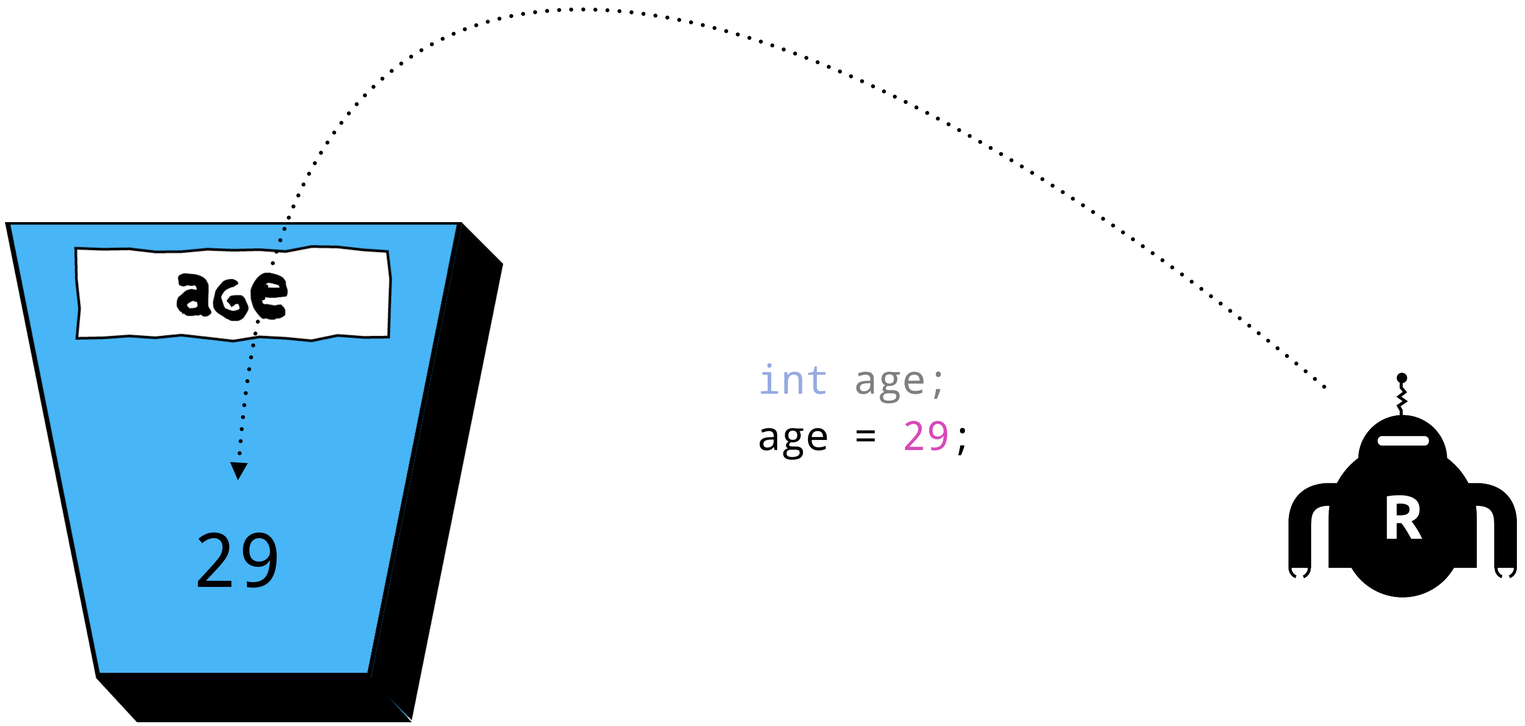
assign a value to a variable




```
int age;
age = 29;
```



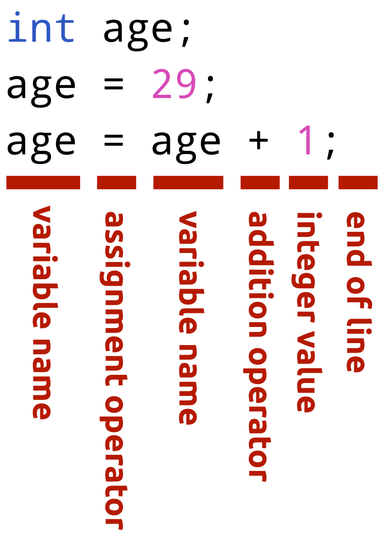
assign a value to a variable



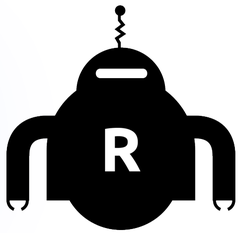
assign a value to a variable

```
int age;
age = 29;
age = age + 1;
```

using a variable

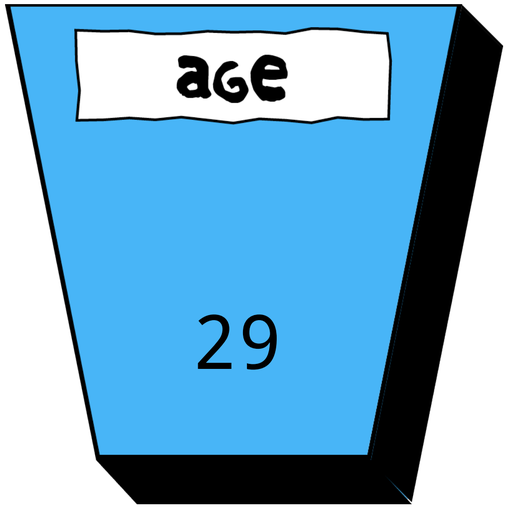


using a variable

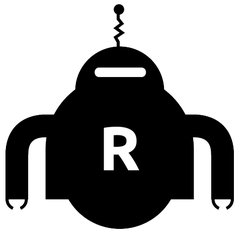



```
int age;
age = 29;
age = age + 1;
```

using a variable

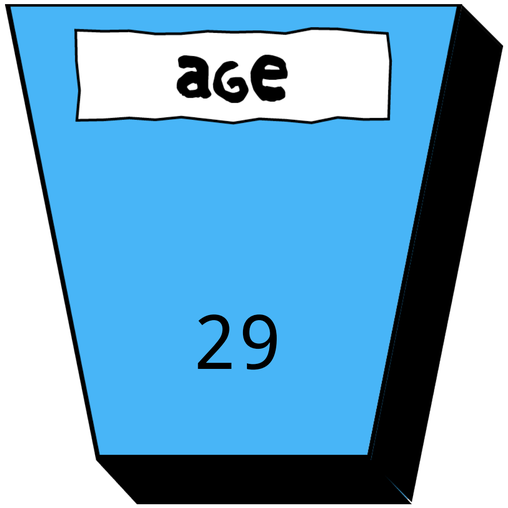



```
int age;
age = 29;
age = age + 1;
```

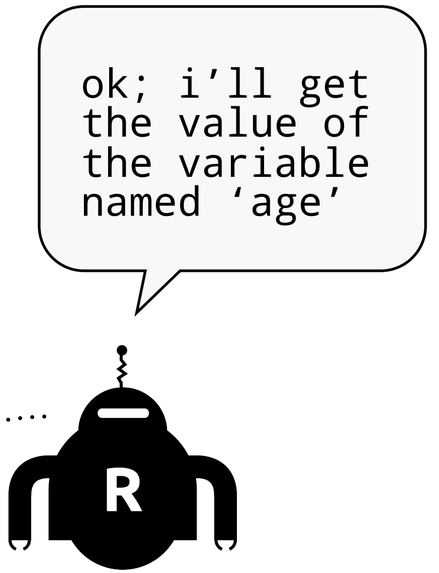


so far i've created an integer variable named age and set it's value to '29'.

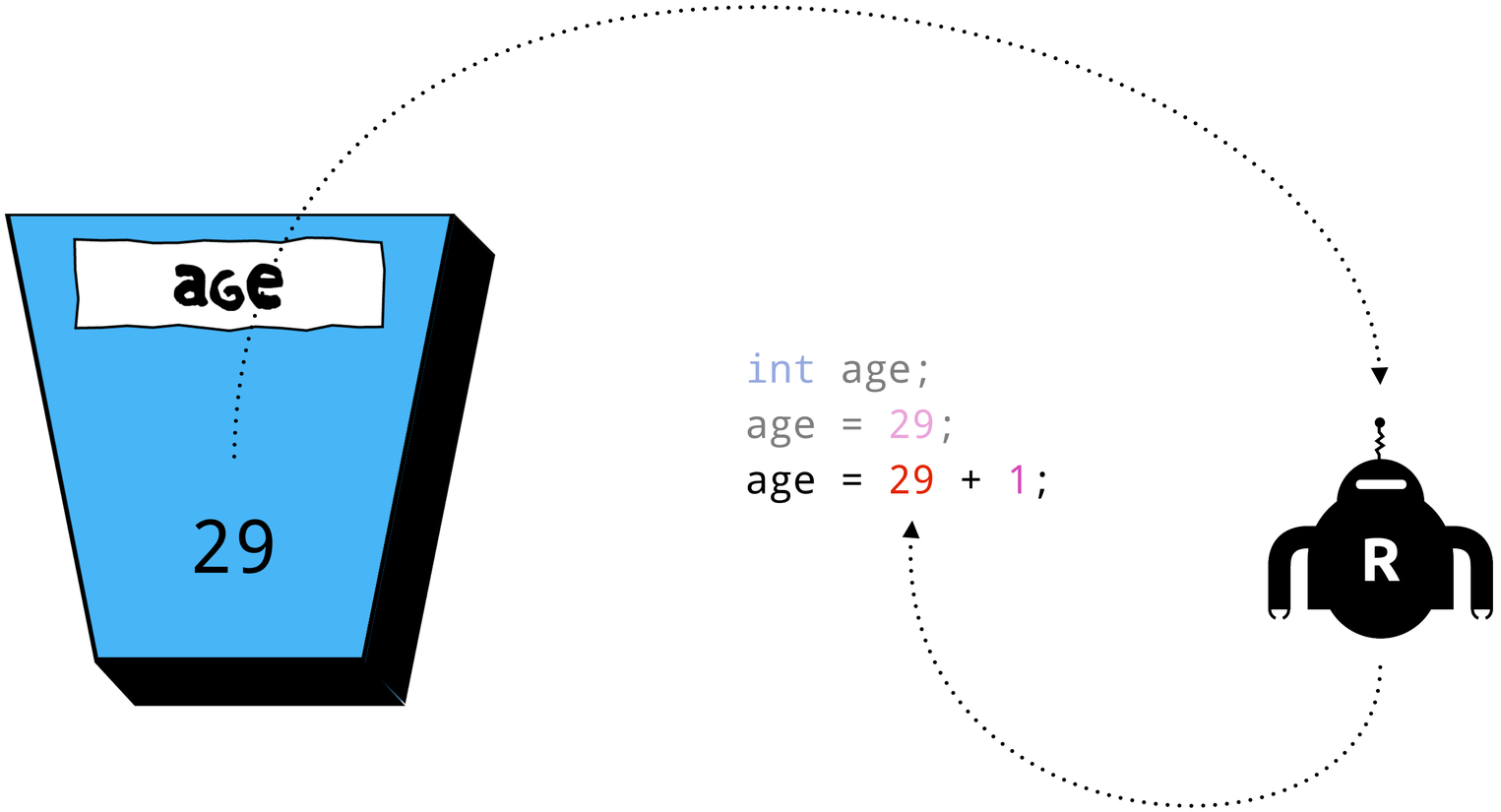
using a variable



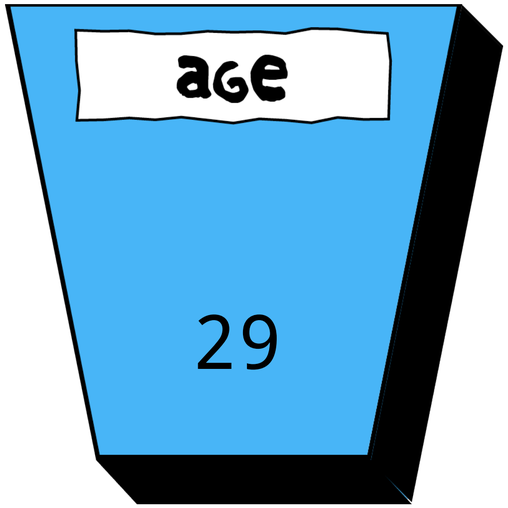

```
int age;
age = 29;
age = age + 1;
```



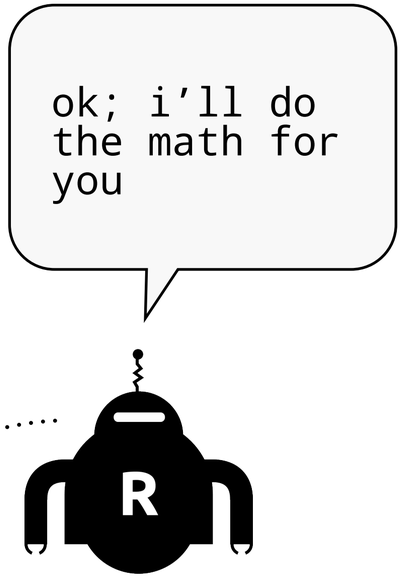
using a variable



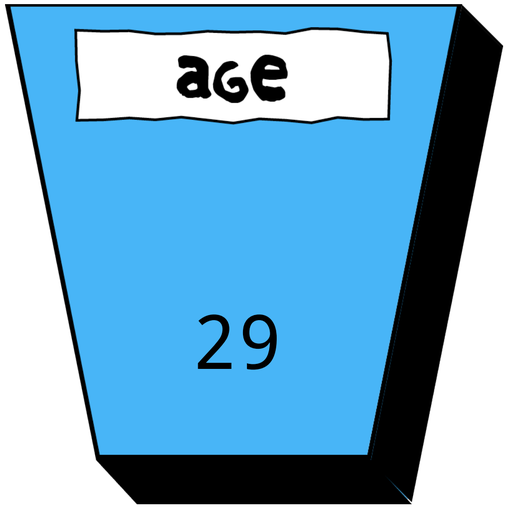
using a variable



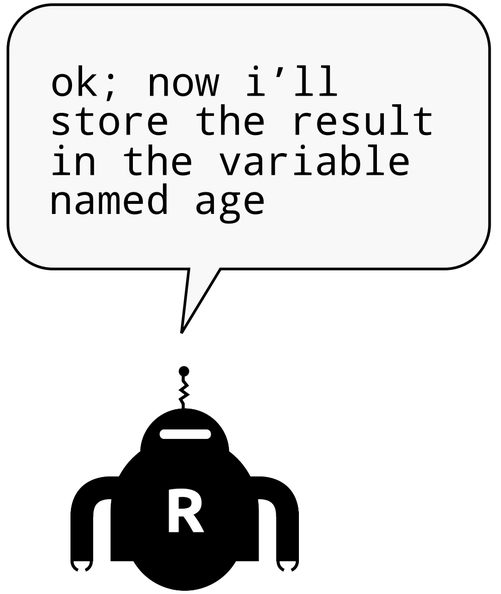

```
int age;
age = 29;
age = 29 + 1;
```



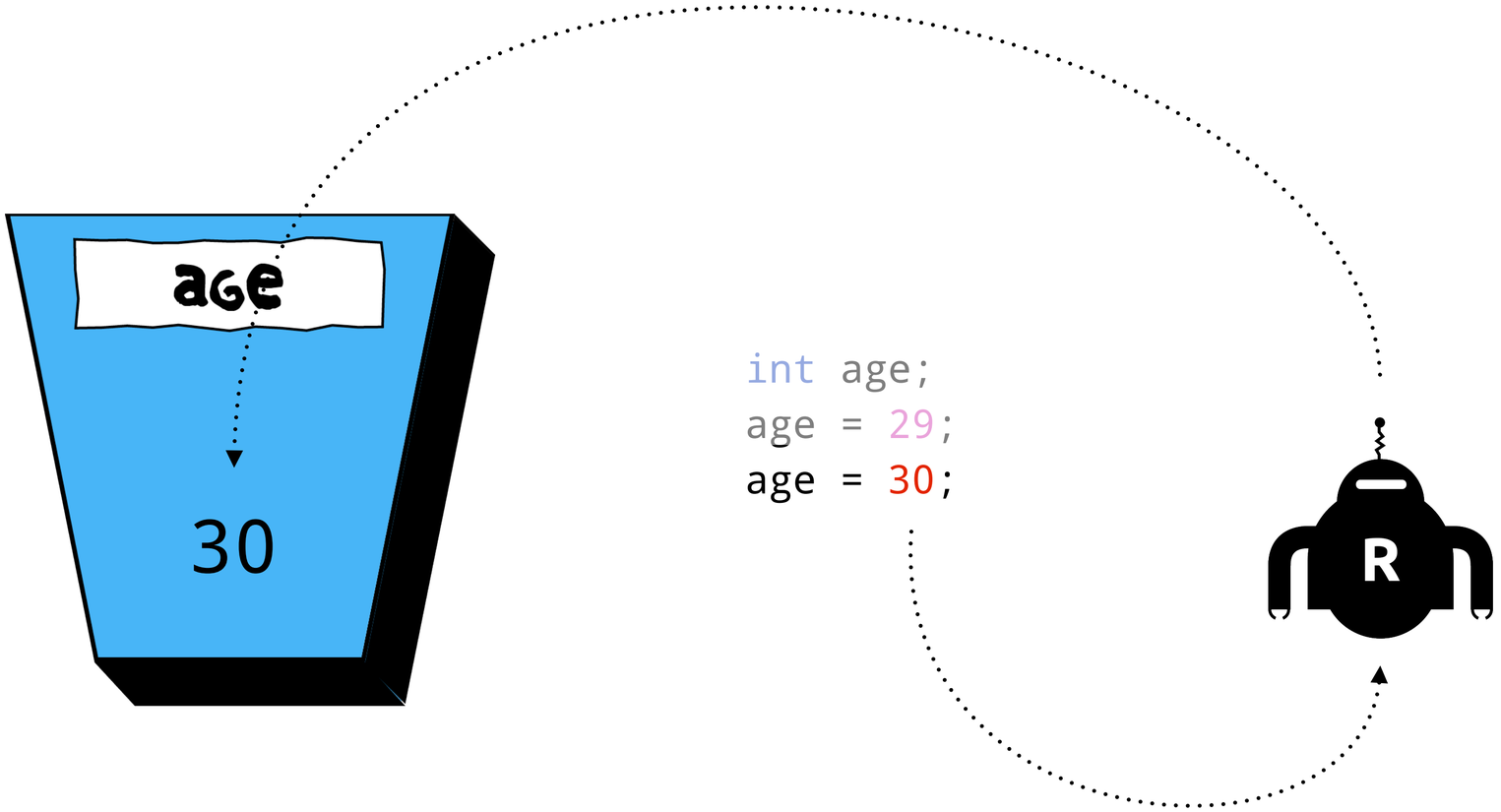
using a variable



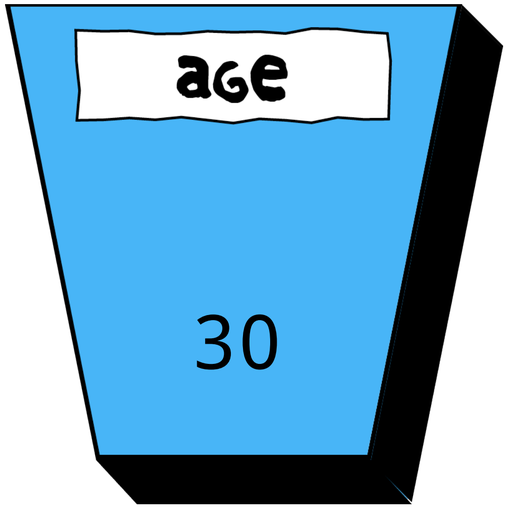

```
int age;
age = 29;
age = 30;
```



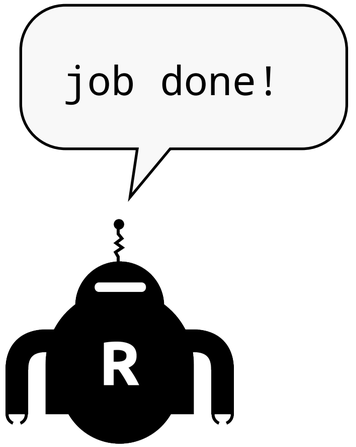
using a variable



using a variable



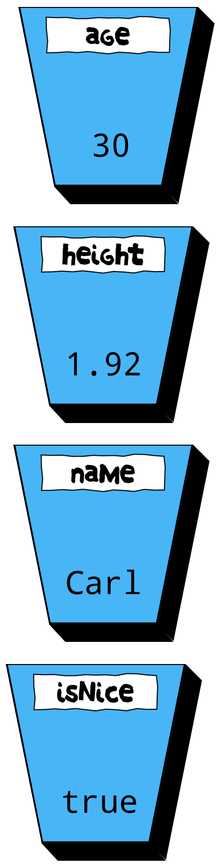

```
int age;
age = 29;
age = age + 1;
```



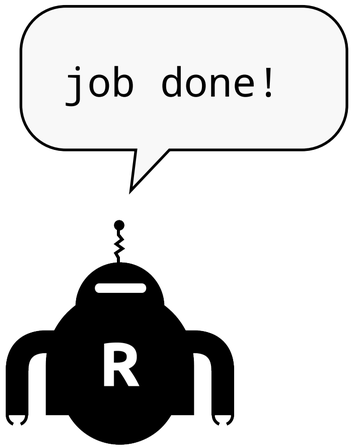
using a variable

```
int age = 30;
double length = 25.4;
String species = ”Blue Whale”;
bool isTagged = true;
```

common data types




```
int age = 30;
double length = 25.4;
String species = ”Blue Whale”;
bool isTagged = true;
```



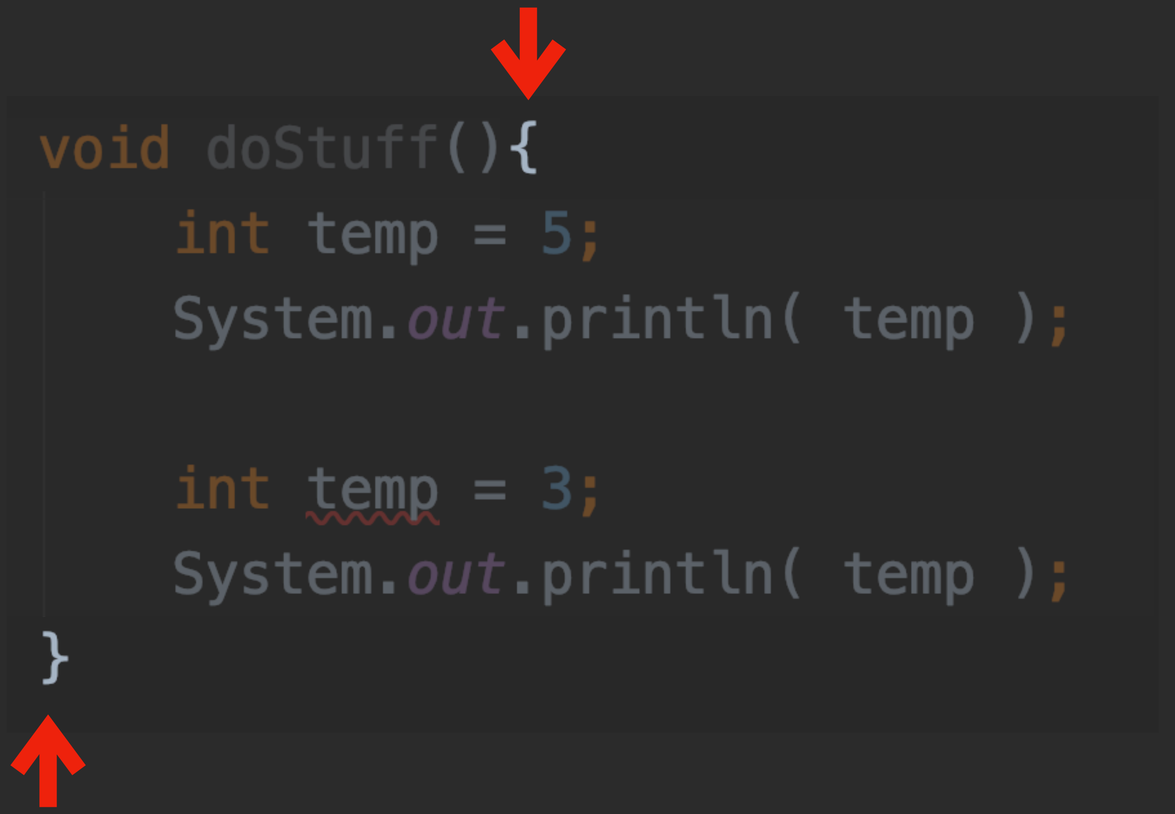
common data types

Variables names has to be **unique** within a scope

```
void doStuff(){
    int temp = 5;
    System.out.println( temp );

    int temp = 3;
    System.out.println( temp );
}
```

This is a **scope**



Fix it

```
void doStuff(){
    int tempA = 5;
    System.out.println( tempA );

    int tempB = 3;
    System.out.println( tempB );
}
```

Same name in separate scope is ok

```
void doA(){
    int temp = 5;
    System.out.println( temp );
}

void doB(){
    int temp = 3;
    System.out.println( temp );
}
```

Variables names cannot be Java keywords

```
void doStuff(){
    int public = 0;
    int static = 0;
    int void = 0;
}
```

Full list of keywords: https://en.wikipedia.org/wiki/List_of_Java_keywords

Primitive datatypes

Boolean, byte, short, int,
long, float, double, char,
and (String)

Primitive datatypes

Internally all datatypes are stored as bits (zeros and ones)

```
104 → 01101000
"hi" →01101000110101
```

Primitive datatypes

Binary value

Boolean	1 bit	true or false

Primitive datatypes

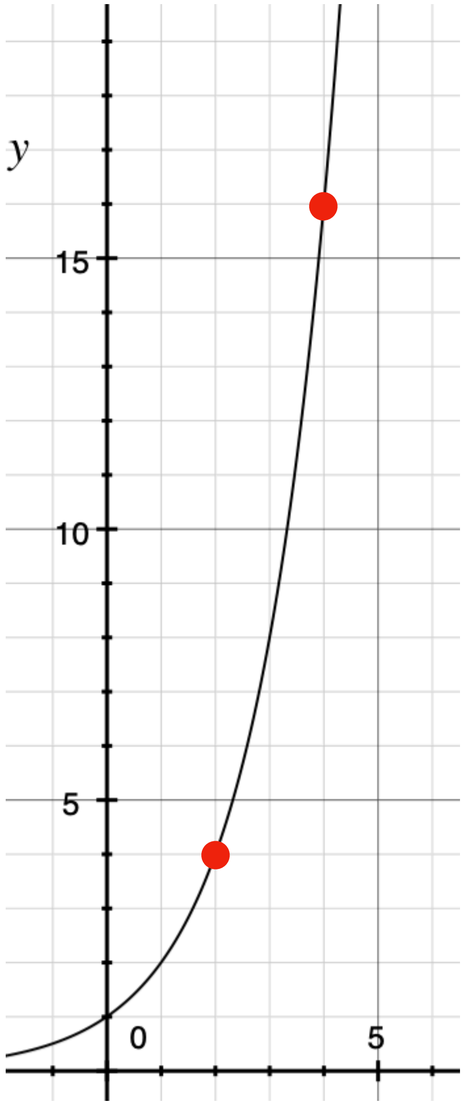
Integers

byte	8 bit	-128 to 127
short	16 bit	-32768 to 32767
int	32 bit	-2147483648 to 2147483647
long	64 bit	-9223372036854775808 to 9223372036854775807

Primitive datatypes

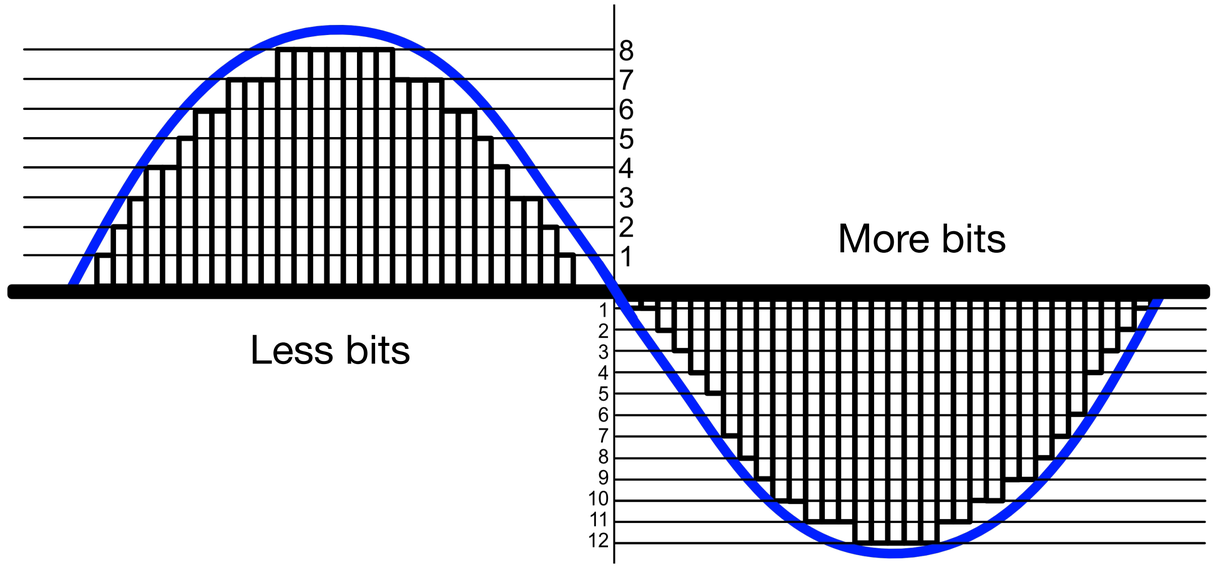
Number of bits and the number of unique values (combinations)

$$y = 2^x$$



Primitive datatypes

Audio CDs are using *short* (16 bit)



Going beyond the max ...

```
static void doStuff(){
    int temp = 2147483647;
    temp = temp + 1;
    System.out.println( temp );
}
```

... wraps to minimum number.

```
static void doStuff(){
    int temp = 2147483647;
    temp = temp + 1;
    System.out.println( temp );
}
```

```
-2147483648
```

Primitive datatypes

Decimals

float	32 bit	6 to 7 decimal digits
double	64 bit	15 decimal digits

Float precision

6-7 digits. Large numbers loose decimal precision.

```
    4.71642
47164.2
```

Float precision

Large scale world positioning issues.

Unity (game engine) positions unreliable beyond 5km.

Float precision

UI positioning using 32-bit float can have precision issues when scrolling far and zooming in.

Primitive datatypes

Characters

char	16 bit	unicode

Characters

In contrast, ASCII characters take only 8 bit

ASCII control characters

00	NULL	(Null character)
01	SOH	(Start of Header)
02	STX	(Start of Text)
03	ETX	(End of Text)
04	EOT	(End of Trans.)
05	ENQ	(Enquiry)
06	ACK	(Acknowledgement)
07	BEL	(Bell)
08	BS	(Backspace)
09	HT	(Horizontal Tab)
10	LF	(Line feed)
11	VT	(Vertical Tab)
12	FF	(Form feed)
13	CR	(Carriage return)
14	SO	(Shift Out)
15	SI	(Shift In)
16	DLE	(Data link escape)
17	DC1	(Device control 1)
18	DC2	(Device control 2)
19	DC3	(Device control 3)
20	DC4	(Device control 4)
21	NAK	(Negative acknowl.)
22	SYN	(Synchronous idle)
23	ETB	(End of trans. block)
24	CAN	(Cancel)
25	EM	(End of medium)
26	SUB	(Substitute)
27	ESC	(Escape)
28	FS	(File separator)
29	GS	(Group separator)
30	RS	(Record separator)
31	US	(Unit separator)
127	DEL	(Delete)

ASCII printable characters

32	space	64	@	96	`
33	!	65	A	97	a
34	"	66	B	98	b
35	#	67	C	99	c
36	$	68	D	100	d
37	%	69	E	101	e
38	&	70	F	102	f
39	'	71	G	103	g
40	(	72	H	104	h
41	)	73	I	105	i
42	*	74	J	106	j
43	+	75	K	107	k
44	,	76	L	108	l
45	-	77	M	109	m
46	.	78	N	110	n
47	/	79	O	111	o
48	0	80	P	112	p
49	1	81	Q	113	q
50	2	82	R	114	r
51	3	83	S	115	s
52	4	84	T	116	t
53	5	85	U	117	u
54	6	86	V	118	v
55	7	87	W	119	w
56	8	88	X	120	x
57	9	89	Y	121	y
58	:	90	Z	122	z
59	;	91	[	123	{
60	<	92	\	124	|
61	=	93	]	125	}
62	>	94	^	126	~
63	?	95	_		

Extended ASCII characters

128	Ç	160	á	192	└	224	Ó
129	ü	161	í	193	┴	225	ß
130	é	162	ó	194	┬	226	Ô
131	â	163	ú	195	├	227	Ò
132	ä	164	ñ	196	─	228	õ
133	à	165	Ñ	197	┼	229	Õ
134	å	166	ª	198	ã	230	µ
135	ç	167	º	199	Ã	231	þ
136	ê	168	¿	200	╚	232	Þ
137	ë	169	®	201	╔	233	Ú
138	è	170	¬	202	╩	234	Û
139	ï	171	½	203	╦	235	Ù
140	î	172	¼	204	╠	236	ý
141	ì	173	¡	205	═	237	Ý
142	Ä	174	«	206	╬	238	¯
143	Å	175	»	207	¤	239	´
144	É	176	░	208	ð	240	≡
145	æ	177	▒	209	Ð	241	±
146	Æ	178	▓	210	Ê	242	‗
147	ô	179	│	211	Ë	243	¾
148	ö	180	┤	212	È	244	¶
149	ò	181	Á	213	ı	245	§
150	û	182	Â	214	Í	246	÷
151	ù	183	À	215	Î	247	¸
152	ÿ	184	©	216	Ï	248	°
153	Ö	185	╣	217	┘	249	¨
154	Ü	186	║	218	┌	250	·
155	ø	187	╗	219	█	251	¹
156	£	188	╝	220	▄	252	³
157	Ø	189	¢	221	¦	253	²
158	×	190	¥	222	Ì	254	■
159	ƒ	191	┐	223	▀	255	nbsp

Primitive datatypes

Text

String	number of bits depends on content