

Syntax

Monday, January 27, 2025 11:18 AM

→ How you write
the code


Expression

↓
generate a value
i.e. $1+1$

"JH" "JB"

Variable → sto

let string : String



String string
String name =

Var a =
var b =

D

A

Reserved word

boolean	abstract	break	class
byte	final	case	extends
char	native	catch	implements
double	private	continue	interface
float	protected	default	throws
int	public	do	
long	static	else	import
short	synchronized	finally	package
void	transient	for	
	volatile	if	instanceof
	strictfp (1.2)	return	new
false	default (1.8)	switch	super
null		throw	this
true		try	
	assert (1.4)	while	
	enum (5.0)		

Names must start
 OK \$ OK _
 after that you

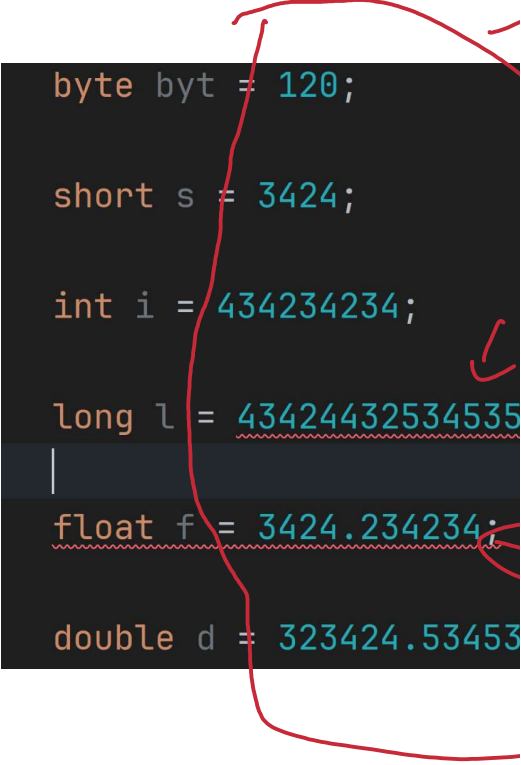
String myFirstName
 ↓
 CamelCase

Pascal Case →

primitive → S

String X → sequ

number X



```
byte byt = 120;

short s = 3424;

int i = 434234234;

long l = 43424432534535

float f = 3424.234234;

double d = 323424.53453
```

Maths:

f

*
/ \rightarrow modulo \rightarrow
0/0

$$\begin{array}{r|l} 7 & 10 \\ \hline & 3 \end{array}$$

(i)
7

/10

7

>
> =

<
< =

= \rightarrow Equivalence

