A+ Computer Science

Number Systems

Number Systems

What is the Standard Base we work with in our everyday lives?

Why do we work in that base?

101010100100000111
100001000011111001
1010101010101011001

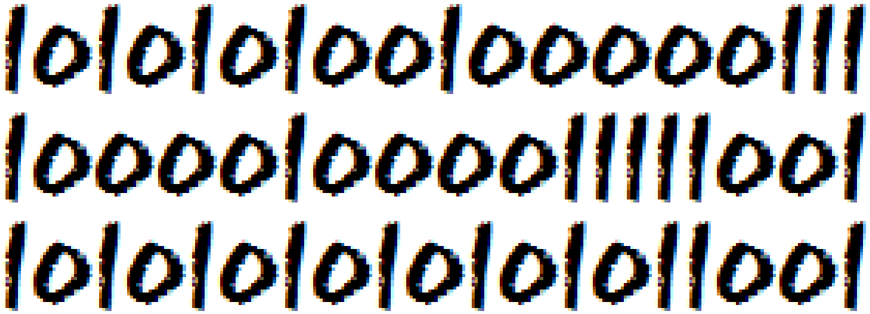
Number Systems

2 – 10100100

8 - 5672

10 – 78645

16 – ABC983EF



bases.java

Base 10

What is 235 really?

Is it 235 or is there more to it?

In actuality, 235 is

2 * 10 to the 2nd power (100) +

3 * 10 to the 1st power (10) +

5 * 10 to the 0th power (1).

If you add these up you end up with 235.

Any base to base 10

All number systems regardless of the base work off of the same principles.

You can convert any base to base 10 by following the power system.

Any base to base 10

Given 32 in base 4, you could convert it by

4^3 4^2 4^1 4^0

* * * *

0 + 0 + 3 + 2

0 * 64 + 0 * 16 + 3 * 4 + 2 * 1

32 in base 4 is 14 in base 10

Base 10 to any base

Given the base 10 number 70, you could convert it to base 5 following these easy steps :

			base to		num10	remainder
1st	divide	70 by 5	5	|	70	0
2nd	divide	14 by 5	5	|	14	4
3rd	divide	2 by 5	5	|	2	2
					0	

The number 70 base 10 = 240 in base 5.

Any base to any base

1st - Convert the number you want to convert to Base 10.

2nd - Convert the Base 10 result to the new base you want.

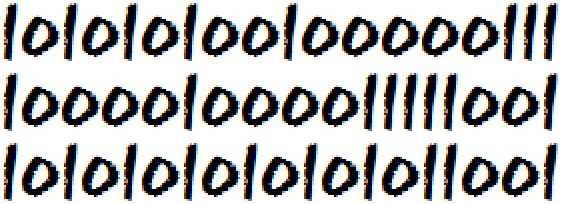
Binary

	Binary digits			
Base 10	8	4	2	1
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0

Base 2 to Base 16

There is a direct conversion from base 2 to base 8 & base 16 without using base 10.

8 and 16 are powers of 2 so they convert directly.



Base 2 to Base 16

Base 2 converts directly to base 16 as each 4 bit section of base 2 equals one base 16 digit.

1111 = 15   15 is maximum single digit for 16

10   11
1010 1011 = AB in base 16

1   4   10
0001 0100 1010 = 14A in base 16

HEX

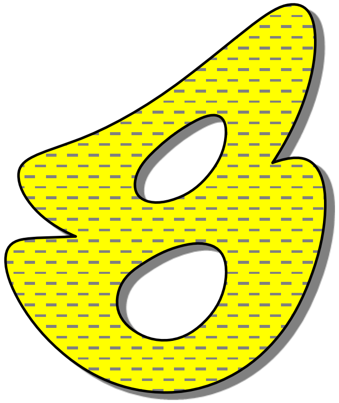
A – 10
B – 11
C – 12
D – 13
E – 14
F – 15

Base 2 to Base 8

Base 2 converts directly to base 16 as each 3 bit section of base 2 equals one base 8 digit.

111 = 7   7 is maximum single digit for base 8

5 3
101 011 = 53 in base 8



1 2 7
001 010 111 = 127 in base 8

Java Base Conversion

```
int base10 = Integer.parseInt("324",6);
out.print("324 base 6 == ");
out.println(base10 + " base10");

out.print("124 base10 == ");
out.println(Integer.toString(base10,16)+" base16\n\n");

out.println(Integer.toBinaryString(90));
out.println(Integer.toOctalString(90));
out.println(Integer.toHexString(90).toUpperCase());
```

OUTPUT

```
324 base 6 == 124 base10
124 base10 == 7c base16


1011010
132
5A
```

javabase.java

Addition Substraction Any Base

Any Base Math

$$\begin{array}{rrl} & 145 & \\ \underline{+} & \underline{345} & \text{base } 8 \\ & 512 & \text{base } 8 \end{array}$$

$$\begin{array}{rrl} & 149 & \\ \underline{+} & \underline{345} & \text{base } 12 \\ & 492 & \text{base } 12 \end{array}$$

$$\begin{array}{rrl} & 427 & \\ - & \underline{345} & \text{base } 9 \\ & 72 & \text{base } 9 \end{array}$$

Bitwise – Binary Operators

& | ^   <<   >>

These operators manipulate the binary digits of variables.

Operator Precedence

()	HIGH
! ++ --	
* / %	
+ -	
<< >> (bitwise shifts)	
< <= > >=	
== !=	
& (bitwise and)	
^ (bitwise xor)	
| (bitwise or)	
&&	
||	
= += -= *= /= %=	
,	LOW

Bitwise AND

```
int one=8;
int two=7;
```

binary representation				
	8	4	2	1
one	1	0	0	0
two	0	1	1	1
result	0	0	0	0

```
out.println("8 & 7 == " + (one&two));
```

OUTPUT

```
8 & 7 == 0
```

Bitwise OR

```
int one=8;
int two=7;
```

binary representation				
	8	4	2	1
one	1	0	0	0
two	0	1	1	1
result	1	1	1	1

```
out.println("8 | 7 == " + (one|two));
```

OUTPUT

```
8 | 7 == 15
```

Bitwise XOR

```
int one=8;
int two=7;
```

binary representation				
	8	4	2	1
one	1	0	0	0
two	0	1	1	1
result	1	1	1	1

```
out.println("8 ^ 7 == " + (one^two));
```

OUTPUT

```
8 ^ 7 == 15
```

bitwiseand.java

bitwiseor.java

bitwisexor.java

Bitwise SHIFT LEFT

int one=8;

16	8	4	2	1
0	1	0	0	0
1	0	0	0	0

```
out.println("8 << 1 == " + (one<<1));
```

SHORTCUT

<< 1   multiplies by 2

OUTPUT

```
8 << 1 == 16
```

Bitwise SHIFT RIGHT

```
int one=8;
```

16	8	4	2	1
0	1	0	0	0
0	0	1	0	0

```
out.println("8 >> 1 == " + (one>>1));
```

SHORTCUT

>> 1   divides by 2

OUTPUT

```
8 >> 1 == 4
```

shiftleft.java
shiftright

Work on Programs!

Crank

Some Code!

A+ Computer Science

Number Systems