Prime Fields : $\mathbb{Z}_p$

In [2]:
```
p:7
```
Out[2]: 7

In [3]:
```
n:1
```
Out[3]: 1

In [4]:
```
gf_set_data(p,n)
```
Out[4]: $\text{Structure }[\text{GF-DATA}]$

In [5]:
```
gf_info()
```
```
characteristic = 7
reduction polynomial = x
primitive element = 3
nr of elements = 7
nr of units = 6
nr of primitive elements = 2
```
Out[5]: **false**

In [6]:
```
gf_add_table()
```
Out[6]:

$$\begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 \\ 1 & 2 & 3 & 4 & 5 & 6 & 0 \\ 2 & 3 & 4 & 5 & 6 & 0 & 1 \\ 3 & 4 & 5 & 6 & 0 & 1 & 2 \\ 4 & 5 & 6 & 0 & 1 & 2 & 3 \\ 5 & 6 & 0 & 1 & 2 & 3 & 4 \\ 6 & 0 & 1 & 2 & 3 & 4 & 5 \end{pmatrix}$$

In [7]:
```
gf_mult_table()
```
Out[7]:

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 2 & 4 & 6 & 1 & 3 & 5 \\ 3 & 6 & 2 & 5 & 1 & 4 \\ 4 & 1 & 5 & 2 & 6 & 3 \\ 5 & 3 & 1 & 6 & 4 & 2 \\ 6 & 5 & 4 & 3 & 2 & 1 \end{pmatrix}$$

If an element α generates with its powers all the elements of the group $\mathbb{Z}_p^*$

then α is a generator of the cyclic multiplicative group and is called **primitive**.

In $\mathbb{Z}_7^*$, 3 and 5 are the **primitive** elements.

In [27]:
```
for i:1 thru p-1 do print(i,gf_order(i))
```
```
1 1
2 3
3 6
4 3
5 6
6 2
```
Out[27]: **done**

Fermat's little theorem

$$p \mid (a^p - a)$$

or, said in another way

$$a^p - a \equiv 0 \mod p$$

In [29]:
```
for a:1 thru p-1 do print(a,mod(a^p-a,p))
```

```
1 0
2 0
3 0
4 0
5 0
6 0
```

Out[29]: **done**

if $a \neq 0$ then we can multiply by a^{-1}

$a^{p-1} - 1 \equiv 0 \mod p$ or $a^{p-1} \equiv 1 \mod p$

In [30]:
```
for a:1 thru p-1 do print(a,mod(a^(p-1),p))
```

```
1 1
2 1
3 1
4 1
5 1
6 1
```

Out[30]: **done**

Inverse computed with Fermat's little theorem

let's multiply again by a^{-1}

$a^{p-2} \equiv a^{-1} \mod p$

In [11]:
```
for a:2 thru p-1 do print(a,"^-1=",mod(a^(p-2),p))
```

```
2 ^-1= 4
3 ^-1= 5
4 ^-1= 2
5 ^-1= 3
6 ^-1= 6
```

Out[11]: **done**

In [12]:
```
gf_primitive()
```

Out[12]: 3

In [16]:
```
gf_make_logs()
```

Out[16]: ["{Lisp Array: \#(1 3 2 6 4 5 1)}" , "{Lisp Array: \#(NIL 0 2 1 4 5 3)}" , "{Lisp Array: \#(2 4 1

In [24]:
```
for j:0 thru p-2 do print(gf_primitive(),"^",j,"=",gf_powers[j])
```

```
3 ^ 0 = 1
3 ^ 1 = 3
3 ^ 2 = 2
3 ^ 3 = 6
3 ^ 4 = 4
3 ^ 5 = 5
```

Out[24]: **done**

In [ ]:
```

```