Hexi’s Wolfram Language Cheat Sheet

Array

Generates a list

```
Array[f, 5]
```

Out[•]=

```
{f[1], f[2], f[3], f[4], f[5]}
```

In[•]:=

```
Array[f, {3, 2}]
```

Out[•]=

```
{{f[1, 1], f[1, 2]}, {f[2, 1], f[2, 2]}, {f[3, 1], f[3, 2]}}
```

Transpose

Takes a list.

Put the nth elements of each list together

In[•]:=

```
Transpose[{{x1, y1, z1}, {x2, y2, z2}, {x3, y3, z3}}]
```

Out[•]=

```
{{x1, x2, x3}, {y1, y2, y3}, {z1, z2, z3}}
```

@: the same as f [x]

@@: the same as apply; replaces the head of the expression

/@: the same as map; applies to each element of the first level

@@@： the same as mapapply; replaces heads at level 1 of the expression by f

In[•]:=

```
f@{a, b, c}
```

Out[•]=

```
f[{a, b, c}]
```

In[•]:=

```
f@@{a, b, c}
```

Out[•]=

```
f[a, b, c]
```

In[•]:=

```
f/@{a, b, c}
```

Out[•]=

```
{f[a], f[b], f[c]}
```

In[•]:=

```
f/@{{a, b}, {c}}
```

Out[•]=

```
{f[{a, b}], f[{c}]}
```

In[•]:=

```
f@@@{{a, b}, {c}}
```

Out[•]=

```
{f[a, b], f[c]}
```

If[condition, t, f, u]

If the condition is true, t; if false, f; if neither, u.

In[ ]:=
```
x = 13.5;
If[x == 12, x, x + 1, 6]
```

Out[ ]=
```
14.5
```