In [1]:

```
!jt -t monokai -T
```

> FIRST THING IS TO INSTALL PULP PACKAGE IN PYTHON

max z 20x1+30x2 s.t x1+2x2<=100 2x1+x2<=100 x1>=0 x2>=0

pulp uses LP solvers(eg GLPK,COINCLP/CBC,CPLEX and GUROBI)

To install pulp in a command prompt,type pip install pulp

In [13]:

```
!pip install pulp
```

```
Requirement already satisfied: pulp in c:\programdata\anaconda3\lib\site-packages (2.7.0)
```

In [14]:

```
#import pulp
# * means all functions are import.
from pulp import *
```

In [15]:

```
# create a lp maximization problem
model=LpProblem("simplex",LpMaximize)
```

In [16]:

```
#pulp.LpVariable(name, lowBound=None, upBound=None, cat='Continuous', e=None
# name – The name of the variable used in the output .lp file
# lowBound – The lower bound on this variable’s range. Default is negative infinity
# upBound – The upper bound on this variable’s range. Default is positive infinity
# cat – The category this variable is in, Integer, Binary or Continuous(default)

x1=LpVariable("x1",lowBound=0,cat="continuous")
x2=LpVariable("x2",lowBound=0,cat="continuous")
```

In [17]:

```
# objective function
model += 20*x1+30*x2
```

In [18]:

```
# constraints
model+= 1*x1+2*x2<=100
model+= 2*x1+1*x2<=100
```

In [19]:

```
# solve with a default solver
model.solve()
```

Out[19]:

```
1
```

In [20]:

```
# print the  solution status
print("status:",LpStatus[model.status])
```

```
status: Optimal
```

In [21]:

```
# show the solution(first approach)
value(x1),value(x2),value(model.objective)
```

Out[21]:

```
(33.333333, 33.333333, 1666.6666500000001)
```

In [22]:

```
# show the solution(second approach)
for v in model.variables():
    print(v.name,"=",v.varValue)
```

```
x1 = 33.333333
x2 = 33.333333
```

In [23]:

```
print("objective :",value(model.objective))
```

```
objective : 1666.6666500000001
```

In [ ]:

```

```