```
from aide_design.play import*

def AWC(root_depth,field_capacity,wilting_point):
  AWC = root_depth * (field_capacity - wilting_point)
  return AWC

root_depth = 2
wilting_point = .15
field_capacity = .30

AWC = AWC(root_depth,field_capacity,wilting_point)

PET = np.array([1.0,2.0,4.0,6.0,8.0,10.0,10.0,8.0,6.0,4.0,2.0,1.0,1.0,2.0])/100
Precip = np.array([109.2,190.5,76.5,39.1,158.0,68.8,61.7,115.3,129.8,65.8,95.3,179.6,79.5,37
AWt = AWC
AW = np.zeros(len(PET)+1)
AW[0] = AWC
recharge = np.zeros(len(PET))
evaporation = np.zeros(len(PET))
P_E = Precip - PET
print(P_E)
for i in range(len(PET)):
  if P_E[i]<0:
    AW[i+1] = AW[i] * np.exp(-PET[i]/AWC)
    recharge[i] = 0
    evaporation[i] = PET[i] * AW[i]/AWC
  else:
    evaporation[i] = PET[i] * AW[i]/AWC
    if AW[i]+P_E[i]<AWC:
      AW[i+1] = AW[i] + P_E[i]
      recharge[i] = 0
    else:
      recharge[i] = AW[i] + P_E[i] - AWC
      AW[i+1]= AW[i]

print(recharge*100)
print(evaporation *100)
```

Month	Precipitation (cm)	PET (cm)	Evaporation (cm)	Recharge (cm)
1/17	10.92	1	1	9.92
2/17	19.05	2	2	17.05
3/17	7.65	4	4	3.65
4/17	3.91	6	6	0
5/17	15.80	8	6.54	2.36
6/17	6.88	10	8.18	0

Month	Precipitation (cm)	PET (cm)	Evaporation (cm)	Recharge (cm)
7/17	6.17	10	5.86	0
8/17	11.53	8	3.36	0
9/17	12.98	6	3.22	0
10/17	6.58	4	3.08	0
11/17	9.53	2	1.71	6.02
12/17	17.96	1	0.85	16.96
1/18	7.95	1	0.85	6.95
2/18	3.711	2	1.71	1.71

Streams are gaining water mostly in the winter and losing water in the summer months.

```
root_depth = 0.5
wilting_point = 0.15
field_capacity = 0.30

AWC = 0.5 * (0.30-.15)

PET = np.array([4.0, 4.0, 4.0, 4.0, 4.0, 4.0, 4.0,4.0, 4.0, 4.0, 4.0, 4.0, 4.0, 4.0])*30/100
precip_2 =np.array([150.2,128.8,177.2,9.6,0.0,0.0,0.0,0.0,0.8,0.2,0.2,12.6,419.0,297.8])/100
recharge = np.zeros(len(PET))
evaporation = np.zeros(len(PET))
AW = np.zeros(len(PET)+1)
AW[0] = AWC
len(recharge)
P_E = precip_2 - PET
for i in range(len(PET)):
  if P_E[i]<0:
    AW[i+1]= AW[i] * np.exp(-PET[i]/AWC)
    recharge[i] = 0
    evaporation[i] = PET[i] * AW[i]/AWC
  else:
    evaporation[i] = PET[i] * AW[i]/AWC
    if AW[i]+P_E[i]<AWC:
      AW[i+1] = AW[i] + P_E[i]
      recharge[i] = 0
    else:
      recharge[i] = AW[i] + P_E[i] - AWC
      AW[i+1]= AWC

print(recharge*100)

plt.plot(precip_2)
plt.plot(evaporation)
```

```
plt.plot(AW[1:])
plt.plot(recharge)
plt.xlabel('Time (months)')
plt.ylabel('Water Depth (m)')
plt.legend(['Precipitation','Evaporation','Available Water','Recharge'])
plt.savefig('groundwater_1')
plt.show()
```

Month	Precipitation (cm)	PET (cm)	Recharge (cm)
1/17	15.02	12	3.02
2/17	12.88	12	0.88
3/17	17.72	12	5.72
4/17	0.96	12	0
5/17	0	12	0
6/17	0	12	0
7/17	0	12	0
8/17	0	12	0
9/17	0.8	12	0
10/17	0.2	12	0
11/17	0.2	12	0
12/17	1.26	12	0
1/18	41.90	12	22.4
2/18	29.78	12	17.78

In the Canning aquifer, the basin is gaining water in the winter which is also the monsoon season and then the rest of the year they are not gaining any water because it is not raining.

```
root_depth = 0.5
wilting_point = 0.15
field_capacity = 0.30

AWC = 0.5 * (0.30-.15)

PET = np.array([4.0, 4.0, 4.0, 4.0, 4.0, 4.0, 4.0,4.0, 4.0, 4.0, 4.0, 4.0])*30/1000
precip_2 =np.array([2.8,11.,6.4,1.9,0.0,34.8,19.2,0.0,0.0,0.0,0.0,37.0])/1000
recharge = np.zeros(len(PET))
evaporation = np.zeros(len(PET))
AW = np.zeros(len(PET)+1)
AW[0] = AWC
len(recharge)
P_E = precip_2 - PET
for i in range(len(PET)):
  if P_E[i]<0:
    AW[i+1]= AW[i] * np.exp(-PET[i]/AWC)
```

```
    recharge[i] = 0
    evaporation[i] = PET[i] * AW[i]/AWC
  else:
    evaporation[i] = PET[i] * AW[i]/AWC
    if AW[i]+P_E[i]<AWC:
      AW[i+1] = AW[i] + P_E[i]
      recharge[i] = 0
    else:
      recharge[i] = AW[i] + P_E[i] - AWC
      AW[i+1]= AWC


print(recharge*100)

plt.plot(precip_2)
plt.plot(PET)
plt.plot(AW[1:])
plt.plot(recharge)
plt.xlabel('Time (months)')
plt.ylabel('Water Depth (m)')
plt.legend(['Precipitation','PET','Available Water','Recharge'])
plt.savefig('groundwater_2')
plt.show()
```