複利法のシミュレーション

お金を預けるとどうなるのか
借りたお金を放置するとどうなるのか

☆重要

預金の複利計算(モデル)

- 複利法
  - 元金に利息を加算したものを
    次の期間の元金として利息計算

- 次期の金額＝現在の金額＋利息
- 利息＝現在の金額×利率

例:10万円を年利5%で預けると

- 以下のモデルを使ってシミュレーション
  - 次期の金額＝現在の金額＋利息
  - 利息＝現在の金額×利率

- Excelでシミュレーション

- Pythonでシミュレーション

例:10万円を年利5%で預けると

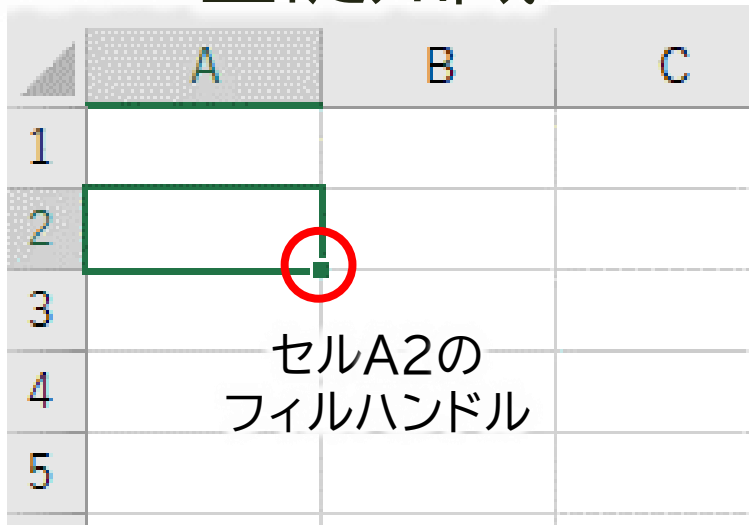
Excelでシミュレーション

- 手順

1. 表の作成
2. 計算式の入力
3. グラフ作成
4. 数値を変えて実行

- 手順の説明は次ページへ

Excelの基礎知識

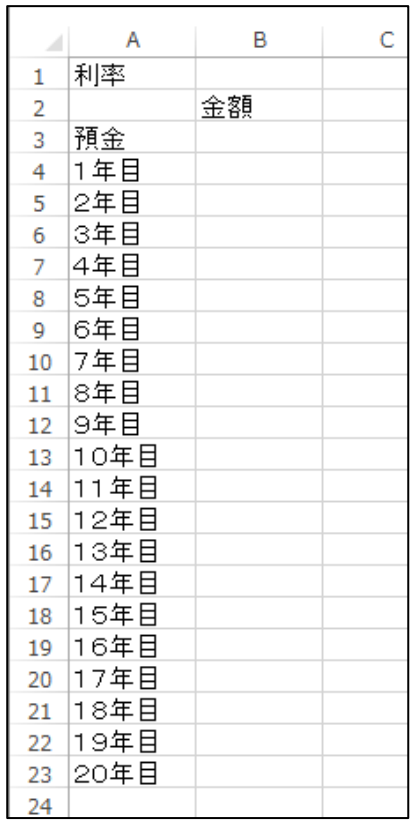


おかしいときは直そうとせず
[Esc]→[Ctrl]+[Z]

例:10万円を年利5%で預けると

Excelでシミュレーション

- 表の作成
  1. セルA1に　利率　と入力
  2. セルB2に　金額　と入力
  3. セルA3に　預金　と入力
  4. セルA4に　1年目　と入力
  5. セルA4のフィルハンドルをドラッグし、
     20年目まで作る



	A	B	C
1	利率		
2		金額	
3	預金		
4	1年目		
5	2年目		
6	3年目		
7	4年目		
8	5年目		
9	6年目		
10	7年目		
11	8年目		
12	9年目		
13	10年目		
14	11年目		
15	12年目		
16	13年目		
17	14年目		
18	15年目		
19	16年目		
20	17年目		
21	18年目		
22	19年目		
23	20年目		
24			

完成イメージ

例:10万円を年利5%で預けると

Excelでシミュレーション

- 計算式の入力
  1. セルB1に　0.05　を入力
  2. セルB3に　100000　を入力
  3. セルB4に　＝B3+INT(B3*$B$1）を入力
  4. セルB4のフィルハンドルをダブルクリック

入力方法(入力する文字はすべて半角)

1. セルB4に　＝　とタイプ
2. セルB3をクリックして　＋　をタイプ
3. int(　とタイプしてからセルB3をクリック
4. ＊　をタイプしてからセルB1をクリック
5. [F4]キーを押す
6.)をタイプしてから[Enter]キーを押す

☆重要

例:10万円を年利5%で預けると

Excelでシミュレーション

- 計算式の入力-計算式の意味
  - 次期の金額＝現在の金額＋利息
  - 利息＝現在の金額×利率

現在の金額　利息

- ＝B3＋INT(B3*$B$1)

整数化　現在の金額　利率

	A	B	C
1	利率	0.05	
2		金額	
3	預金	100000	
4	1年目	=B3+INT(B3*$B$1)	
5	2年目	1 1	

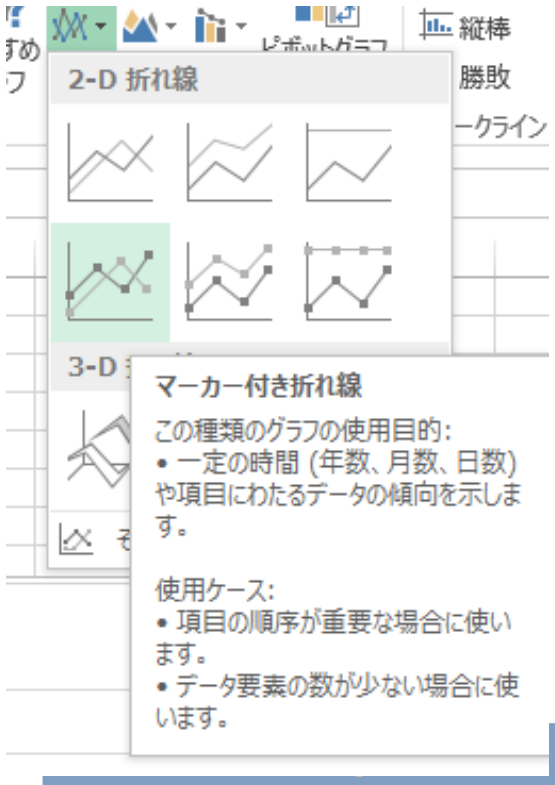
INT(数値)

☆重要

例:10万円を年利5%で預けると

Excelでシミュレーション

- グラフ作成
  1. セルA2からセルB23までドラッグ
  2. [挿入]→[折れ線グラフ]とクリック
  3. [マーカー付折れ線]をクリック



- グラフから150,000円を超えるのは何年後か読み取ってみよう

☆重要

例:10万円を年利5%で預けると

Excelでシミュレーション

- 数値を変えて実行
  - 現在の実際の利率は0.1%以下です
  - 0.1%の利率で利息部分が1000円を超える（101000になる）のは何年後かシミュレーションする

- 自分のドライブに
  fukuri という名前で保存

例:10万円を年利5%で預けると

Pythonでシミュレーション

- 手順

1. プログラムの作成・実行
2. グラフ作成のために改造・実行
3. グラフ表示プログラムの追加・実行
4. 数値を変えて実行

- 手順の説明は次ページへ

例:10万円を年利5%で預けると

Pythonでシミュレーション

- プログラムの作成・実行
  1. 新しいノートブックを用意
  2. 以下のプログラムを入力
  3. 入力できたら実行
  4. 20年目まで表示するよう改造

```
yokin=100000
riritsu=0.05
for i in range(10):
    risoku=int(yokin*riritsu)
    yokin=yokin+risoku
    print(i+1,'年目:',yokin)
```

☆重要

例:10万円を年利5%で預けると

Pythonでシミュレーション

- プログラムの作成・実行-プログラムの意味

```
yokin=100000
riritsu=0.05
for i in range(10):
    risoku=int(yokin*riritsu)
    yokin=yokin+risoku
    print(i+1,'年目:',yokin)
```

```
#変数 yokin に 100000 を代入
#変数 riritsu に 0.05 を代入
#変数 i を0～9に変える
    #変数 risoku に yokin*riritsu を整数にして代入
    #変数 yokin に yokin+risoku を代入
    # i+1,'年目:',yokin  を表示
```

- 次期の金額＝現在の金額＋利息
- 利息＝現在の金額×利率

例:10万円を年利5%で預けると

Pythonでシミュレーション

- グラフ作成のために改造・実行
  - 以下の部分を修正して実行し動作確認

```
import matplotlib.pyplot as plt
yokin=[100000]
riritsu=0.05
for i in range(20):
    risoku=int(yokin[i]*riritsu)
    yokin.append(yokin[i]+risoku)
    print(i+1,'年目:',yokin[i+1])
```

例:10万円を年利5%で預けると

Pythonでシミュレーション

- グラフ表示プログラムの追加・実行
  - 以下の部分を追加して実行し動作確認

```
plt.title('fukuri')
plt.xlabel('Year')
plt.ylabel('Yokin')
plt.plot(yokin,marker='o')
plt.grid(True)
plt.show()
```

```
import matplotlib.pyplot as plt
yokin=[100000]
riritsu=0.05
for i in range(20):
    risoku=int(yokin[i]*riritsu)
    yokin.append(yokin[i]+risoku)
    print(i+1,'年目:',yokin[i+1])

plt.title('fukuri')
plt.xlabel('Year')
plt.ylabel('Yokin')
plt.plot(yokin,marker='o')
plt.grid(True)
plt.show()
```

完成イメージ

☆重要

例:10万円を年利5%で預けると

Pythonでシミュレーション

- 数値を変えて実行
  - 現在の実際の利率は0.1%以下です
  - 0.1%の利率で利息部分が1000円を超える（101000になる）のは何年後かシミュレーションする

生命体の増加シミュレーション

☆重要

生命体の増加シミュレーション(モデル)

- ある空間の中の生命体の増加について
  - 個体が多ければ増加する
  - 個体数が多すぎると減少する
- 増加数＝個体数×増加率
- 減少率＝(個体数/環境収容数)×増加率
- 減少数＝個体数×減少率

例:個体数10,増加率0.01,環境収容数1000

- 以下のモデルを使ってシミュレーション
  - 増加数＝個体数×増加率
  - 減少率＝(個体数/環境収容数)×増加率
  - 減少数＝個体数×減少率

- Excelでシミュレーション

- Pythonでシミュレーション

例:個体数10,増加率0.01,環境収容数1000

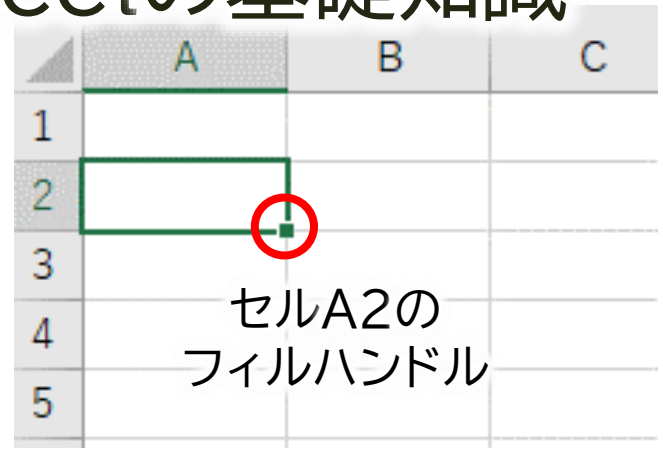
Excelでシミュレーション

- 手順

1. 表の作成
2. 計算式の入力
3. グラフ作成
4. 効率の良いタイミングを調べる

- 手順の説明は次ページへ

Excelの基礎知識

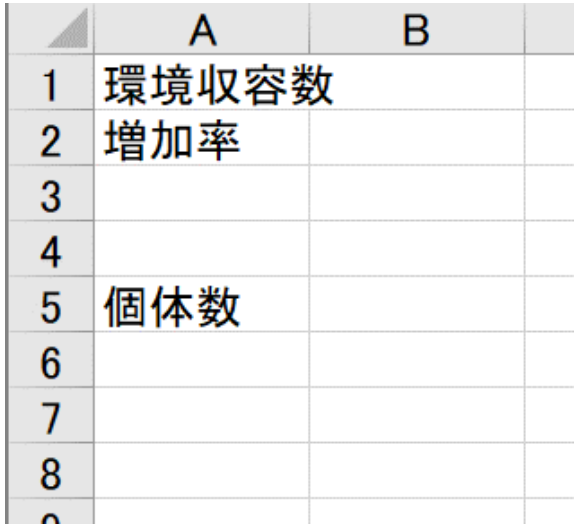


おかしいときは直そうとせず
[Esc]→[Ctrl]+[Z]

例:個体数10,増加率0.01,環境収容数1000

Excelでシミュレーション

- 1.表の作成
  1. セルA1に　環境収容数　と入力
  2. セルA2に　増加率　と入力
  3. セルA5に　個体数　と入力



	A	B
1	環境収容数	
2	増加率	
3		
4		
5	個体数	
6		
7		
8		

完成イメージ

例:個体数10,増加率0.01,環境収容数1000

Excelでシミュレーション

- 2.計算式の入力
  1. セルB1に　1000　を入力
  2. セルB2に　0.01　を入力
  3. セルA6に　10　を入力
  4. セルA7に　=A6+A6*$B$2-A6*(A6/$B$1)*$B$2　を入力
  5. セルA7のフィルハンドルを下にドラッグしてセルA1006までドラッグする

☆重要

例:個体数10,増加率0.01,環境収容数1000

Excelでシミュレーション

- 2.計算式の入力-計算式の意味
  - 増加数＝個体数×増加率
  - 減少率＝(個体数/環境収容数)×増加率
  - 減少数＝個体数×減少率

現在の数　増加数　減少数

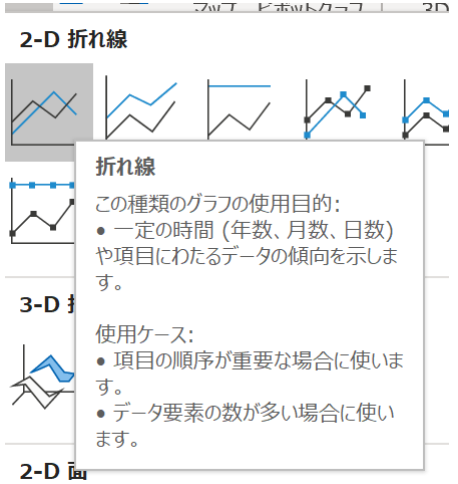
- ＝A6+A6*$B$2-A6*(A6/$B$1)*$B$2

	A	B	C	D
1	環境収容数	1000		
2	増加率	0.01		
3				
4				
5	個体数			
6	10			
7	=A6+A6*$B$2-A6*(A6/$B$1)*$B$2			
8	10.19897			

例:個体数10,増加率0.01,環境収容数1000

Excelでシミュレーション

- 3.グラフ作成
  1. セルA5からセルA1006までドラッグ
  2. [挿入]→[-折れ線グラフ]とクリック
  3. [折れ線]をクリック



☆重要

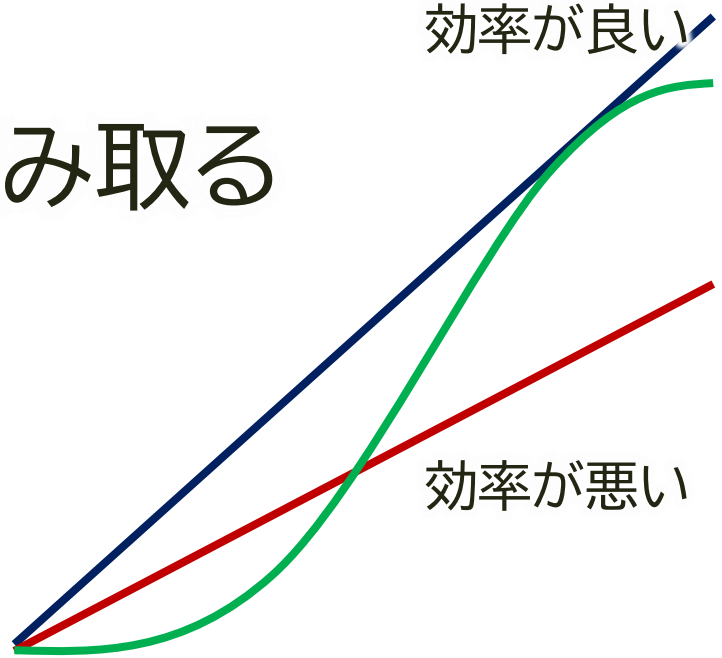
例:個体数10,増加率0.01,環境収容数1000

Excelでシミュレーション

- 4.効率の良いタイミングを調べる
- グラフ上の点と原点を結ぶ直線の傾きが最大
  →増加の効率が一番高い

- 一番効率が良い時間をグラフから読み取る

- 自分のドライブに
  zouka という名前で保存



例:個体数10,増加率0.01,環境収容数1000

Pythonでシミュレーション

- 手順

1. プログラムの作成・実行
2. グラフ作成のために改造・実行
3. グラフ表示プログラムの追加・実行
4. 効率の良いタイミングを調べる

- 手順の説明は次ページへ

例:個体数10,増加率0.01,環境収容数1000

Pythonでシミュレーション

- プログラムの作成・実行
  1. 新しいノートブックを用意
  2. 以下のプログラムを入力
  3. 入力できたら実行

```
zouka=0.01
capacity=1000
n=10
for  i  in range(1000):
    zoukasuu=n*zouka
    gensyousuu=n*(n/capacity)*zouka
    n=(n+(zoukasuu - gensyousuu))
    print(n)
```

☆重要

例:個体数10,増加率0.01,環境収容数1000

Pythonでシミュレーション

- プログラムの作成・実行-プログラムの意味

```
zouka=0.01
capacity=1000
n=10
for  i  in range(1000):
    zoukasuu=n*zouka
    gensyousuu=n*(n/capacity)*zouka
    n=(n+(zoukasuu - gensyousuu))
    print(n)
```

```
# 増加率
# 環境収容数
# 最初の個体数

# 増加数
# 減少数
# 個体数
```

- 増加数＝個体数×増加率
- 減少率＝(個体数/環境収容数)×増加率
- 減少数＝個体数×減少率

例:個体数10,増加率0.01,環境収容数1000

Pythonでシミュレーション

- 2.グラフ作成のために改造・実行
  - 以下の部分を修正して実行し動作確認

```
import matplotlib.pyplot as plt
zouka=0.01
capacity=1000
n=[10]
for i in range(1000):
    zoukasuu=n[i]*zouka
    gensyousuu=n[i]*(n[i]/capacity)*zouka
    n.append(n[i]+(zoukasuu - gensyousuu))
    print(n[i+1])
```

例:個体数10,増加率0.01,環境収容数1000

Pythonでシミュレーション

- 3.グラフ表示プログラムの追加・実行
  - 以下の部分を追加して実行し動作確認

```
plt.plot(n)
plt.title("number of life")
plt.xlabel("time")
plt.ylabel("number")
plt.show()
```

```
import matplotlib.pyplot as plt
zouka=0.01
capacity=1000
n=[10]
for i in range(1000):
    zoukasuu=n[i]*zouka
    gensyousuu=n[i]*(n[i]/capacity)*zouka
    n.append(n[i]+(zoukasuu-gensyousuu))
    print(n[i+1])

plt.plot(n)
plt.title("number of life")
plt.xlabel("time")
plt.ylabel("number")
plt.show()
```

完成イメージ

☆重要

例:個体数10,増加率0.01,環境収容数1000

Pythonでシミュレーション

- 4.効率の良いタイミングを調べる
- グラフ上の点と原点を結ぶ直線の傾きが最大
  →増加の効率が一番高い

- 一番効率が良い時間をグラフから読み取る