Compound interest

- Say you invest $100 and make 7% annually
- After one year you have:
  \[ \text{\$100 + \$100 \times 7/100 = \$107} \]
  You made \$7.00
- If you leave it invested, and make another 7% the next year, you have:
  \[ \text{\$107 + \$107 \times 7/100 = \$114.48} \]
  You made \$7.49

Compound interest

- The more you have the more you make
  
  \[ \$100 \text{ at 7% Annual interest} \]
  Doubles every seven years.

Debt and Payments

- Example debt:
  - Beginning balance \$1000
  - Interest rate 13% annually
  - Make a payment of \$300 per year
- After one year you owe:
  \[ \text{\$1000 + \$1000 \times 13/100 = \$1130 (\$130 in interest)} \]
- Then you pay \$300, so you owe \$1130 - \$300 = \$870
- The next year, you owe:
  \[ \text{\$870 + \$870 \times 13/100 = 983.10 (\$113.10 in interest)} \]
- Pay another \$300, and get down to \$683.10

New Assignment

- Given a debt amount, interest rate and monthly payment, figure out how long it takes to pay off the debt, and how much you end up paying in interest.

“Compounded monthly”

- Instead of computing and adding interest every year, do it every month.
- Use interest rate of \((13 / 12)\% = 1.0833\% \text{ every month)}
- Is 13% compounded monthly better than 13% compounded annually? Is it exactly the same?
- We'll write a program to figure this out, using the main technical feature we'll need in the compound interest program – the while loop.
## While loop

- Do a block of statements a bunch of times.
- Stop when something changes.

```python
while Boolean is True:
    Do statements in block
```

- `while` loop does block over and over, until the Boolean expression becomes False somehow.

## Infinite loop

- One of the classic programming bugs
- Get out of it using CRTL-c (hold down control key and type c)
- Repeat after me: CRTL-c

## Ways a program can fail

- Crash: gets to a statement Python cannot execute, program stops and prints red error messages.
- Does something, but not what it is supposed to do.
- Infinite loop: gets stuck doing the same thing over and over, will never escape.
- Often when a program "goes away" or "hangs up" it is in an infinite loop.

## EIR program

- Study this program on the course Web page.

## Uglier code, prettier output

- You can ask Python to cut off extra decimal places.

```python
'%.2f'.format(1.8976437521)
```

- `format()` is a function
- The string describing the specific format is stuck onto the beginning, with the period inbetween
- The `format` function produces a string
- This is an expression, of type string.
## Challenging Assignment

- If you understand this, you are home free in this course.
- Start the program early. Go to section.
- You may work with a partner on this assignment! Make sure you both understand what is going on.