repitisjon

Mål

- Dokumentasjon
- Testing
- optimalisering

Git

- Versjonskontroll.
- Gå tilbake til tidligere versjoner.
- Bra for å jobbe sammen.

workflow

- github - lage nyt repo
- git clone
- git add
- git add
- git commit

Så jobbe

- Sjekke status
- git push
- git pull
- Conflicts
- checkout
- git .gitignore
    - Ignorerer filer.
- ...

Python

- Property - @property
    - Decorator
    - Kun lesbare
    - setters - @<function name>.setter
- Masgiske metoder
    - init, call, str, add, sub, mul, eq, repr, ...

- Statick methods
    - funksjoner som ikke tar inn self.
    - disse funksjonene kan kalles uten å lage en instans.
- fire pillarer i OOP.
    1. abstraksjon
        - Program to an interface not an implementation"
        - Skriv tester først.
    2. Innkapsling
        - Gjøre funksjoner private.
    3. Polymorfisme
        - funksjoner som gjør forkjselige ting bassert på input
    4. Arv
        - class(parent)
        - super().method()
        - når: er en forhold (is a relationship)
            - A dog is an animal
        - Når skal man ikke bruke arv:
            - har en forhold

Design patterns

- Erfarne programmerere som har laget en oppsrift på vanlige problemer.
- Ex
    - favor object composition over class inheritance

Kodestil

- Python har pep8
    - Mellorrom mellom argumenter
- verktøy:
    - flake8, black, atuopep8

Documentasjon

- Dockstring
    - i python """ """
- Kan generere documentasjon ut i fra dockstring.

Pålitlelig kode

- Assert for å sjekke at alt er ok
- Bra å feile tidlig

- Raise exception
    - forskjellige typer errors

Hvorfor burde man skrive tester

- Det tar mye tid å skrive tester, men man burde skrive tester for å sjekke programmet. det må jo testes læl.
- Om koden skal kjøres i en bedrift burde all kode være testet.

Typer tester

- Enhetstester
    - Tester som tester små enheter i isolasjon
- Integrasjonstester
    - Tester hvor man tester ulike enheter fungerer sammen.
- Regresjontester
    - ..

Vi kan bruke pytest for å kjøre alle funksjoner som starter med test.

C++

- dtypes:
    - int, float
- referanse, og pekere
- const for å forhindre endring av en variabel.
- Struct, and class

```
struct Point{
    double x
}

class Circle{
    private:
    double radius
    Point center;
    public:
    Cirlce(double r){
        radius = r;
        ...
    }
}
```

- Stacken og heapen
    - Må huske å dealokere minne.

Datastruktuerer

- Dynamiske array

    - En liste med en gitt kapasitet blir allokert
- Lenkede lister
    - Elementer i lista er noder
- Binary search Tree
    - Rekursive
    - rot node, løv blader.