



< P...log... (1)>



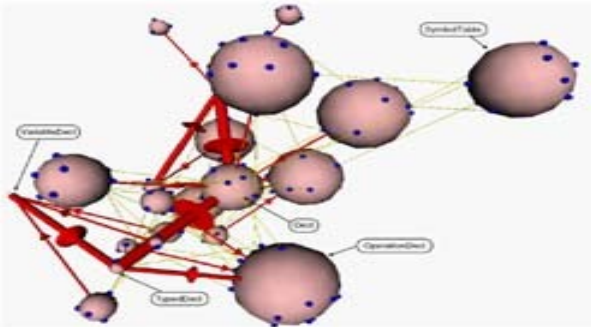
< → P<sub>on</sub> D<sub>a</sub> Mod<sub>l</sub> >



The Python Data Model

Prologue

< Funct...s • o...s (3)>



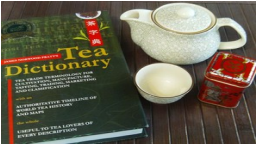
Functions as objects

< • a<sub>y</sub> • seq...es >



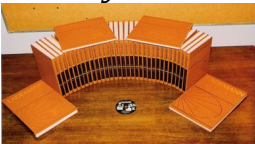
An array of sequences

< Dict...s • sets >



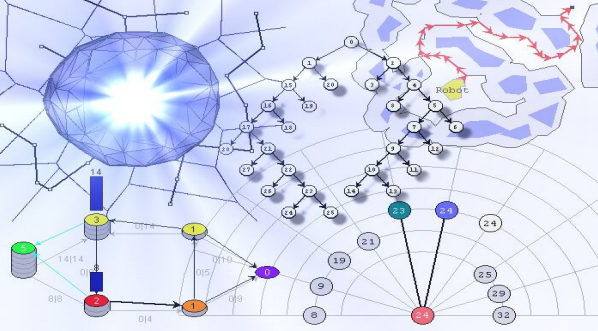
Dictionaries and sets

< T<sub>xt</sub> vs. bytes >



Text versus bytes

< D struct...s (3)>



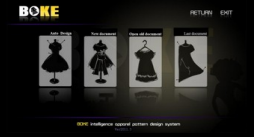
Data Structures

< First-c<sub>as</sub> f<sub>ns</sub> >



First-class functions

< D<sub>sn</sub> p<sub>at</sub>terns • 'f<sub>cf</sub>'-s >



Design patterns with first-class functions

< Fn deco...rs • cl...s >



Function decorators and closures

< O<sub>bj</sub> ref...s, m<sub>ut</sub> & r<sub>ec</sub> >



Object references, mutability and recycling

< • Pythonic o<sub>bj</sub> >



A Pythonic object

< Seq...e h<sub>ack</sub>, h<sub>ash</sub> & s<sub>lic</sub> >



Sequence hacking, hashing and slicing

< I...faces: • p<sub>ro</sub> • ABCs >



Interfaces: from protocols to ABCs

< I-herit...e: • g<sub>ood</sub> • • w<sub>orse</sub> >



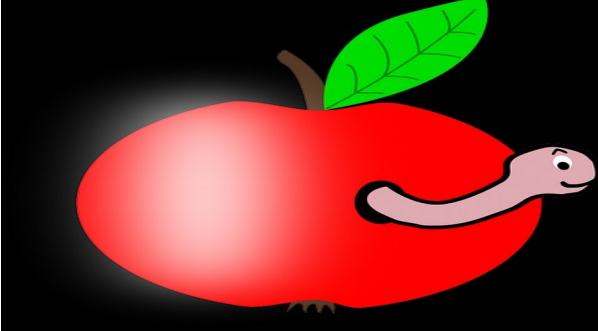
Inheritance: for good or for worse

< Op...r ov<sub>er</sub>g: d • r >



Operator overloading: doing it right

< O O Id...ms (6)>



Object Oriented Idioms

< C...flow (5)>



Control flow

< Itera...s, i & g >



Iterables, iterators and generators

< Coro...s >



Coroutines

< Con...t m<sub>an</sub> & e b >



Context managers and else blocks

< C...cy • fut...s >



Concurrency with futures

< C...cy • asyn-o >



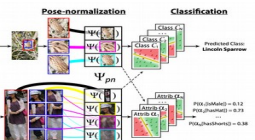
Concurrency with asyncio

< Dyna-c a & p >



Dynamic attributes and properties

< a descri...s >



Attribute descriptors

< Class-m<sub>et</sub>p<sub>rog</sub> >



Class metaprogramming

< Meta...g (3)>



Metaprograming