Corso ITS: *ARTIFICIAL INTELLIGENCE SPECIALIST*

Modulo: **Programmazione ad oggetti in Python e librerie esterne**

Docente: *Andrea Ribuoli*

Giovedì **6 Marzo** 2025

09:00 - 14:00

---

integrale.py

- immaginate gli assi cartesiani **x** (*ascisse*) e **y** (*ordinate*)
- suddividete il segmento che va da 0 a 1 sull'asse delle ascisse in 100 parti uguali
- considerate i 100 rettangoli che hanno per base i 100 segmenti e
- per altezza il valore `sqrt(1 - x ** 2)`
- dove x è la posizione iniziale di ciascuna micro-base lungo l'asse *x*
- sommate le aree dei 100 rettangoli così determinati
- dividete il valore determinato per quello di **pi greco**
- generalizzate il programma in modo da rendere parametrico
- il numero di parti in cui dividere il segmento 0-1
- quanto vale il rapporto con pi greco impostando 1000 parti?
- e 10000?
- Sapete darvi una spiegazione?

In [1]:
```
1 / 100
```

Out[1]:
```
0.01
```

In [8]:
```
from math import sqrt, pi
b = 1 / 100
area = 0.0
for i in range(0, 100) :
    area += b * sqrt(1 - (b * i) ** 2)
print(area / pi)
```

```
0.25149799641972553
```

In [1]:
```
from math import sqrt, pi
def integra(parti) :
    b = 1 / parti
    area = 0.0
    for i in range(0, parti) :
        area += b * sqrt(1 - (b * i) ** 2)
    return area / pi
```

In [2]:
```
print(integra(100))
```

```
0.25149799641972553
```

In [3]:
```
print(integra(1000))
```

```
0.2501561957212206
```

In [4]:
```
print(integra(10000))
```

```
0.25001582191291544
```

area del cerchio è pari a πr^2

- con il calcolo fatto stiamo approssimando un quarto dell'area del cerchio
- ecco perchè all'aumentare del frazionamento in parti sempre più piccole
- la differenza dal valore esatto si riduce

In [5]:
```
print(integra(1000000))
```

```
0.2500001590613622
```

In [6]:
```
print(integra(10000000))
```

```
0.25000001591251536
```

- un esempio (non efficiente) per consentire all'utente di specificare una funzione:
- - questa verrà ancora integrata tra 0 e 1
- - determino `y(x)` in due tempi:
- - prima formatto la stringa costante sostituendo i riferimenti `{x}` con il valore `b*i`
- - poi sottometto la funzione `eval()` che determinerà **y**

In [19]:
```
# user_expr = input("Indica una funzione di x racchiudendo x tra parentesi g
USER_EXPR = "10 - {x} ** 2 + {x} * 25"
```

In [26]:
```
def integra(parti) :
    b = 1 / parti
    area = 0.0
    for i in range(0, parti) :
        area += b * (eval(USER_EXPR.format(x = b*i)))
    return area / pi
```

In [31]:
```
print(integra(100000))
```

```
7.0558309465490945
```

SQLAlchemy

- pip3 install SQLAlchemy
- pip3 install psycopg2

In [36]:
```
import sqlalchemy
sqlalchemy.__version__
```

Out[36]:
```
'1.4.32'
```

In [2]:
```
import psycopg2
psycopg2.__version__
```

Out[2]:
```
'2.9.1 (dt dec pq3 ext lo64)'
```

In [ ]:
```
from sqlalchemy import create_engine
engine = create_engine(
    "postgresql+psycopg2://postgres:papapa@localhost/italy"
    )
```

In [22]:
```
from sqlalchemy import MetaData
metadata_obj = MetaData()
```

In [23]:
```
from sqlalchemy import Table, Column, Integer, String
user_table = Table(
     "user_account",
     metadata_obj,
     Column("id", Integer, primary_key=True),
     Column("name", String(30)),
     Column("fullname", String),
 )
```

In [31]:
```
from sqlalchemy import Table, Column, Integer, String
province = Table(
     "provincia",
     metadata_obj,
     Column("sigla", String(2), primary_key=True),
     Column("nome", String(30)),
     Column("residenti", Integer),
```

```
 )
```

In [25]:
```
metadata_obj.create_all(engine)
```

In [26]:
```
!echo "\dt\q" | PGPASSWORD=papapa psql -U postgres italy
```

```
           List of relations
 Schema |     Name     | Type  |  Owner
--------+--------------+-------+----------
 public | provincia    | table | postgres
 public | user_account | table | postgres
(2 rows)
```

In [11]:
```
!echo "\d user_account\q" | PGPASSWORD=papapa psql -U postgres italy
```

```
                                    Table "public.user_account"
  Column  |         Type          | Collation | Nullable |                 D
efault
----------+-----------------------+-----------+----------
+----------------------------------------
 id       | integer               |           | not null | nextval('user_acc
ount_id_seq'::regclass)
 name     | character varying(30) |           |          |
 fullname | character varying     |           |          |
Indexes:
    "user_account_pkey" PRIMARY KEY, btree (id)
```

In [27]:
```
!echo "\d provincia\q" | PGPASSWORD=papapa psql -U postgres italy
```

```
                     Table "public.provincia"
  Column   |         Type          | Collation | Nullable | Default
-----------+-----------------------+-----------+----------+---------
 sigla     | character varying(2)  |           | not null |
 nome      | character varying(30) |           |          |
 residenti | integer               |           |          |
Indexes:
    "provincia_pkey" PRIMARY KEY, btree (sigla)
```

---

psycopg2 only

In [37]:
```
import psycopg2
conn = psycopg2.connect(database="italy",
                        host="localhost",
                        user="postgres",
                        password="papapa",
                        port="5432")
```

In [38]:
```
cursor = conn.cursor()
```

In [50]:
```
cursor.execute("INSERT INTO provincia (sigla, nome) VALUES('PU', 'Pesaro e U
```

In [51]:
```
cursor.execute("COMMIT TRANSACTION")
```

In [52]:
```
!echo "SELECT * FROM provincia;\q" | PGPASSWORD=papapa psql -U postgres ital
```

```
 sigla |      nome       | residenti
-------+-----------------+-----------
 PU    | Pesaro e Urbino |
(1 row)
```

In [69]:
```
!echo "DELETE FROM provincia;\q" | PGPASSWORD=papapa psql -U postgres italy
```

```
DELETE 110
```

In [71]:
```
import psycopg2
conn = psycopg2.connect(database="italy",
                        host="localhost",
                        user="postgres",
                        password="papapa",
                        port="5432")
from csv import reader
infile = open("province.csv")
csvReader = reader(infile)
header = True
for riga in csvReader :
    if header :
        header = False
        continue
    sigla = riga[0]
    nome = riga[1].replace("'", "''")
    residenti = int(riga[2])
    cursor.execute(f"INSERT INTO provincia (sigla, nome, residenti) VALUES('
infile.close()
cursor.execute("COMMIT TRANSACTION")
```

In [72]:
```
!echo "SELECT * FROM provincia;\q" | PGPASSWORD=papapa psql -U postgres ital
```

sigla	nome	residenti
RM	Roma	4353738
MI	Milano	3218201
NA	Napoli	3107006
TO	Torino	2277857
PA	Palermo	1268217
BS	Brescia	1262318
BA	Bari	1260142
CT	Catania	1113303
BG	Bergamo	1109933
SA	Salerno	1104731
FI	Firenze	1014423
BO	Bologna	1009210
PD	Padova	936274
CE	Caserta	924166
VR	Verona	921557
VA	Varese	890043
TV	Treviso	885972
MB	Monza e della Brianza	868859
VI	Vicenza	865082
VE	Venezia	854275
GE	Genova	850071
LE	Lecce	802082
CS	Cosenza	711739
MO	Modena	700862
PG	Perugia	660690
ME	Messina	636653
FG	Foggia	628556
CO	Como	600190
CN	Cuneo	589108
TA	Taranto	583479
LT	Latina	574891
CA	Cagliari	560373
RC	Reggio Calabria	553861
PV	Pavia	547251
TN	Trento	538604
RE	Reggio Emilia	532483
UD	Udine	531466
BZ	Bolzano	524256
FR	Frosinone	493067
AN	Ancona	474124
PR	Parma	448899
AG	Agrigento	442049
TP	Trapani	434476
AL	Alessandria	426658
AV	Avellino	423506
PI	Pisa	421851
MN	Mantova	412610
SR	Siracusa	402822
BR	Brindisi	397083
FC	Forlì-Cesena	394067
BT	Barletta-Andria-Trani	392546
RA	Ravenna	391414
LU	Lucca	390042
CH	Chieti	389169

PZ	Potenza	370680
NO	Novara	370143
CZ	Catanzaro	362343
PU	Pesaro e Urbino	360711
CR	Cremona	359388
FE	Ferrara	348362
AR	Arezzo	344374
LC	Lecco	339238
LI	Livorno	337334
RN	Rimini	336786
SS	Sassari	333116
RG	Ragusa	321359
PE	Pescara	321309
VT	Viterbo	319008
MC	Macerata	318921
PN	Pordenone	312051
TE	Teramo	309859
AQ	L'Aquila	301910
PT	Pistoia	291839
PC	Piacenza	286758
BN	Benevento	279675
SV	Savona	279408
CL	Caltanissetta	269710
SI	Siena	268341
PO	Prato	254608
RO	Rovigo	238588
TS	Trieste	234682
LO	Lodi	229338
TR	Terni	228218
CB	Campobasso	224644
GR	Grosseto	223045
SP	La Spezia	220698
AT	Asti	216677
IM	Imperia	215130
AP	Ascoli Piceno	209450
BL	Belluno	205781
MT	Matera	199685
MS	Massa-Carrara	196580
SO	Sondrio	181437
BI	Biella	178551
KR	Crotone	175566
FM	Fermo	174849
VC	Vercelli	173868
EN	Enna	168052
VV	Vibo Valentia	161619
OR	Oristano	160746
OT	Olbia-Tempio	160672
VB	Verbano-Cusio-Ossola	159664
RI	Rieti	157420
NU	Nuoro	156096
GO	Gorizia	139673
AO	Aosta	126883
CI	Carbonia-Iglesias	126324
VS	Medio Campidano	98623
IS	Isernia	85805
OG	Ogliastra	57185

```
(110 rows)
```