project2024-1-benoit

Version

Table of Contents

Contents:

project2024-1-benoit documentation

Add your content using `reStructuredText` syntax. See the reStructuredText documentation for details.

python-1-2024

Oefeningen package

Submodules

Oefeningen.Kalender module

Oefeningen.Kalender. calculate_day_of_week (*day* , *month* , *year*) [source]

Calculate day of the week

Parameters :

- day
- month
- year

Return type :

day_of_week

Oefeningen.Kalender. days_in_month (*month_to_display : int* , *year : int*) → int [source]

Get number of days in month

Parameters :

- month_to_display
- year

Returns :

int

Oefeningen.Kalender. draw (* , *days_to_print : int* , *first_day_to_print : int* , *month_to_print : int* , *year_to_print : int*) → None [source]

Oefeningen.Kalender. is_leap_year (*year : int*) → bool [source]

Calculate if year is a leap year :param year:

Returns :

bool

Oefeningen.Kalender. main ( ) [source]

Oefeningen.lijsten module

Oefeningen.lijsten. add_type (*upper : str*) [source]

Oefeningen.lijsten. count_words (*string*) [source]

Oefeningen.sortlist module

Module contents

chapter1 package

Module contents

chapter2 package

Submodules

chapter2.2e_ex_2_2_3 module

chapter2.ex2 module

chapter2.ex2. leap_year (*year*) [source]

chapter2.ex2. main ( ) [source]

chapter2.ex_3_3_chatgtp module

chapter2.ex_3_3_chatgtp. draw_grid (*columns* , *rows* , *box_size*) [source]

chapter2.ex_3_3_full module

chapter2.ex_3_3_full. edge_line (*horizontal* , *size*) [source]

chapter2.ex_3_3_full. squares (*horizontal* , *vertical* , *size*) [source]

chapter2.ex_3_3_full. vol_line (*horizontal* , *size*) [source]

chapter2.ex_3_3_v1 module

chapter2.ex_3_3_v1. boxes ( ) [source]

chapter2.ex_3_3_v1. do_four (*f* , *p*) [source]

chapter2.ex_3_3_v1. do_twice (*f*) [source]

chapter2.ex_3_3_v1. horizontal_line ( ) [source]

chapter2.ex_3_3_v1. print_char (*c*) [source]

chapter2.ex_3_3_v1. vertical_line (_) [source]

chapter2.ex_3_3_v2 module

chapter2.ex_3_3_v2. boxes ( ) [source]

chapter2.ex_3_3_v2. horizontal_line ( ) [source]

chapter2.ex_3_3_v2. horizontal_part (_) [source]

chapter2.ex_3_3_v2. n_times (*n* , *f* , *p*) [source]

chapter2.ex_3_3_v2. print_char (*c*) [source]

chapter2.ex_3_3_v2. vertical_line (_) [source]

chapter2.ex_3_3_v2. vertical_part (_) [source]

chapter2.ex_3_3_v3 module

chapter2.ex_3_3_v3. boxes ( ) [source]

chapter2.ex_3_3_v3. line (*c* , *f* , * *p*) [source]

chapter2.ex_3_3_v3. n_times (*n* , *f* , * *p*) [source]

chapter2.ex_3_3_v3. part (*c* , *e*) [source]

chapter2.ex_3_3_v3. print_char (*c*) [source]

chapter2.ex_3_print_right module

chapter2.ex_3_print_right. main ( ) [source]

chapter2.ex_3_print_right. print_right (*text*) [source]

chapter2.ex_3_triangle module

Module contents

chapter3 package

Submodules

chapter3.ex3 module

chapter3.ex3. draw_grid (*row_count* , *col_count* , *inner_width*) [source]

chapter3.ex3. fac1 (*n : int*) → int [source]

Parameters :

n (*object*)

chapter3.ex3. fact (*n*) [source]

chapter3.ex3. is_valid_iban (*iban*) [source]

Validate a IBAN string

Parameters :

- iban (*str*) – IBAN string
- Returns – bool: True if valid, False otherwise

chapter3.ex3. leap_year (*year : int*) → bool [source]

Parameters :

year (*object*)

Returns: bool

chapter3.ex3. main_start ( ) [source]

chapter3.ex_3_3_chatgtp module

chapter3.ex_3_3_chatgtp. draw_grid (*columns* , *rows* , *box_size*) [source]

chapter3.ex_3_3_full module

chapter3.ex_3_3_full. edge_line (*horizontal* , *size*) [source]

chapter3.ex_3_3_full. squares (*horizontal* , *vertical* , *size*) [source]

chapter3.ex_3_3_full. vol_line (*horizontal* , *size*) [source]

chapter3.ex_3_3_v1 module

chapter3.ex_3_3_v1. boxes ( ) [source]

chapter3.ex_3_3_v1. do_four (*f* , *p*) [source]

chapter3.ex_3_3_v1. do_twice (*f*) [source]

chapter3.ex_3_3_v1. horizontal_line ( ) [source]

chapter3.ex_3_3_v1. print_char (*c*) [source]

chapter3.ex_3_3_v1. vertical_line (_) [source]

chapter3.ex_3_3_v2 module

chapter3.ex_3_3_v2. boxes ( ) [source]

chapter3.ex_3_3_v2. horizontal_line ( ) [source]

chapter3.ex_3_3_v2. horizontal_part (_) [source]

chapter3.ex_3_3_v2. n_times (*n* , *f* , *p*) [source]

chapter3.ex_3_3_v2. print_char (*c*) [source]

chapter3.ex_3_3_v2. vertical_line (_) [source]

chapter3.ex_3_3_v2. vertical_part (_) [source]

chapter3.ex_3_3_v3 module

chapter3.ex_3_3_v3. boxes ( ) [source]

chapter3.ex_3_3_v3. line (*c* , *f* , * *p*) [source]

chapter3.ex_3_3_v3. n_times (*n* , *f* , * *p*) [source]

chapter3.ex_3_3_v3. part (*c* , *e*) [source]

chapter3.ex_3_3_v3. print_char (*c*) [source]

chapter3.ex_3_print_right module

chapter3.ex_3_print_right. main ( ) [source]

chapter3.ex_3_print_right. print_right (*text*) [source]

chapter3.ex_3_triangle module

chapter3.functions_demo module

chapter3.functions_demo. calculate_gcd (*a : int* , *b : int*) → int [source]

Calculate the greatest common divisor of a and b :param a: :type a: object :param b: :type b: object

Returns:int

chapter3.functions_demo. calculate_gcd_alt (*a* , *b*) [source]

chapter3.start module

chapter3.test_ex3 module

class chapter3.test_ex3. TestIbanValidation (*methodName = 'runTest'*) [source]

Bases: `TestCase`

test_invalid_iban_checksum ( ) [source]
test_invalid_iban_length ( ) [source]
test_invalid_iban_structure ( ) [source]
test_valid_iban ( ) [source]
test_valid_iban_with_spaces ( ) [source]

Module contents

chapter4 package

Submodules

chapter4.beue_klant module

chapter4.comment module

chapter4.ex4 module

chapter4.ex4. arc (*radius : float* , *angle : float*) [source]

chapter4.ex4. circle (*tur: <module 'turtle' from 'C:\\Users\\benoit\\.conda\\envs\\syntra\\Lib\\turtle.py'>* , *radius: int*) [source]

chapter4.ex4. pentagon (*tur: <module 'turtle' from 'C:\\Users\\benoit\\.conda\\envs\\syntra\\Lib\\turtle.py'>* , *length: float*) [source]

chapter4.ex4. polygon (*t: <module 'turtle' from 'C:\\Users\\benoit\\.conda\\envs\\syntra\\Lib\\turtle.py'>* , *lenght: float* , *n*) [source]

chapter4.ex4. square (*t: <module 'turtle' from 'C:\\Users\\benoit\\.conda\\envs\\syntra\\Lib\\turtle.py'>* , *lenght: int*) [source]

chapter4.ex4. start ( ) [source]

chapter4.ex4-turtles module

chapter4.fac module

chapter4.fac. fac1 (*n : int*) → int [source]

chapter4.fac. fac2 (*n : int*) → int [source]

Calculate the factorial of a non-negative integer *n* using an iterative approach.

The factorial of a non-negative integer n, denoted as n!, is the product of all positive integers less than or equal to n. By definition, the factorial of 0 and 1 is 1.

Parameters :

n (*int*) – A non-negative integer whose factorial is to be calculated.

Returns :

The factorial of the input integer *n* .

Return type :

int

Raises :

ValueError – If *n* is a negative integer.

Examples

fac2(5) -> 120

chapter4.fac. fac3 (*n : int*) → int [source]

chapter4.fac. fac4 (*n : int*) → int [source]

chapter4.fac. main ( ) [source]

chapter4.jupyturtle module

jupyturtle.py release 2024-03 Celebrating Think Python Third Edition”

class chapter4.jupyturtle. Turtle (* , *auto_render = True* , *delay : float | None = None* , *drawing : Drawing | None = None*) [source]

Bases: `object`

back (*units : float*) [source]

Move the turtle backward by units, drawing if the pen is down.

property **delay**

forward (*units : float*) [source]

Move turtle forward by units; leave trail if pen is down.

get_SVG ( ) [source]

property **heading** *: float*

hide ( ) [source]

Hide turtle. It will still leave trail if the pen is down.

jumpto (*x : float* , *y : float*) [source]

Teleport the turtle to coordinates (x, y) without drawing.

left (*degrees : float*) [source]

Turn turtle left by degrees.

moveto (*x : float* , *y : float*) [source]

Move the turtle to coordinates (x, y), drawing if the pen is down.

pendown ( ) [source]

Lower the pen, so turtle starts drawing.

penup ( ) [source]

Lift the pen, so turtle stops drawing.

render ( ) [source]

right (*degrees : float*) [source]

Turn turtle right by degrees.

show ( ) [source]

Show turtle.

property **x** *: float*

property **y** *: float*

chapter4.jupyturtle. back (* *args*)

Move the turtle backward by units, drawing if the pen is down.

chapter4.jupyturtle. bk (** args*)

Move the turtle backward by units, drawing if the pen is down.

chapter4.jupyturtle. fd (** args*)

Move turtle forward by units; leave trail if pen is down.

chapter4.jupyturtle. forward (** args*)

Move turtle forward by units; leave trail if pen is down.

chapter4.jupyturtle. get_turtle ( ) → Turtle [source]

Gets existing _main_turtle; makes one if needed.

chapter4.jupyturtle. hide (** args*)

Hide turtle. It will still leave trail if the pen is down.

chapter4.jupyturtle. jumpto (** args*)

Teleport the turtle to coordinates (x, y) without drawing.

chapter4.jupyturtle. left (** args*)

Turn turtle left by degrees.

chapter4.jupyturtle. lt (** args*)

Turn turtle left by degrees.

chapter4.jupyturtle. make_turtle (*** , *auto_render = True* , *delay = None* , *width = 300* , *height = 150*) → None [source]

Makes new Turtle and sets _main_turtle.

chapter4.jupyturtle. moveto (** args*)

Move the turtle to coordinates (x, y), drawing if the pen is down.

chapter4.jupyturtle. pendown (** args*)

Lower the pen, so turtle starts drawing.

chapter4.jupyturtle. penup (** args*)

Lift the pen, so turtle stops drawing.

chapter4.jupyturtle. render (** args*)

chapter4.jupyturtle. right (** args*)

Turn turtle right by degrees.

chapter4.jupyturtle. rt (** args*)

Turn turtle right by degrees.

chapter4.jupyturtle. show (** args*)

Show turtle.

chapter4.thinkpython module

chapter4.void module

chapter4.void. toto (*a* , *b*) [source]

Module contents

chapter5 package

Submodules

chapter5.ex5 module

chapter5.ex5. calculate_number_of_days ( ) [source]

chapter5.ex5. check_fermat (*a* , *b* , *c* , *n*) [source]

chapter5.ex5. is_triangle (*param* , *param1* , *param2*) [source]

chapter5.ex5. main ( ) [source]

chapter5.hanoi_sol module

chapter5.hanoi_sol. hanoi_tower (*n* , *src* , *helper* , *dest*) [source]

chapter5.hanoi_sol. main ( ) [source]

chapter5.rec_search module

chapter5.rec_search. main ( ) [source]

chapter5.rec_search. search_file (*directory* , *target_file*) [source]

chapter5.recusions_demo module

chapter5.recusions_demo. even_nbr (*n*) [source]

chapter5.recusions_demo. fac3 (*n*) [source]

Calculate the factorial of a positive n. :param n: int: The number to calculate the factorial for.

Returns :

The factorial of a positive n.

Return type :

int

chapter5.recusions_demo. factorial_me (*n : int*) [source]

Calculate the factorial of a positive n. :param n: int: The number to calculate the factorial for.

Returns :

The factorial of a positive n.

Return type :

int

chapter5.recusions_demo. new_fac (*n : int*) → int [source]

Calculate the factorial of a positive n. :param n: int: The number to calculate the factorial for.

Returns :

The factorial of a positive n.

Return type :

int

chapter5.recusions_demo. old_fac (*n : int*) → int [source]

Bereken de faculteit van een gegeven geheel getal.

De functie gebruikt een iteratief proces om de faculteit van ‘n’ te berekenen. Als n kleiner is dan 2, retourneert de functie 1.

Parameters: n (int): Het geheel getal waarvan de faculteit moet worden berekend.

Returns: int: De faculteit van ‘n’. Als n kleiner is dan 2, retourneert de functie 1.

Raises :

ValueError – Als ‘n’ een negatief getal is.

chapter5.recusions_demo. sum_n (*n*) [source]

Module contents

chapter6 package

Submodules

chapter6.ex6 module

chapter6.ex6. ackermann (*m* , *n*) [source]

chapter6.ex6. gcd (*a* , *b*) [source]

chapter6.ex6. hypot (*a* , *b*) [source]

chapter6.ex6. is_between (*x* , *y* , *z*) [source]

chapter6.ex6. is_power (*a* , *b*) [source]

Module contents

chapter7 package

Submodules

chapter7.ex-7 module

Module contents

main module

main. main ( ) [source]

main. rectangle (*height* , *width_rec* , *star*) [source]