

Boat buddy

API Documentation

December 5, 2019

Contents

Contents	1
1 Module WGS84DecToDeg	2
1.1 Functions	2
1.2 Variables	2
2 Module WGS84DegToDec	3
2.1 Functions	3
2.2 Variables	3
3 Module angleToFunction	4
3.1 Functions	4
3.2 Variables	4
4 Module functionIntersect	5
4.1 Functions	5
4.2 Variables	5
5 Module map	6
5.1 Class DrawMap	6
5.1.1 Methods	6
5.2 Class Arrow	9
5.2.1 Methods	9
6 Module mareeCalculator	10
6.1 Functions	10
6.2 Variables	10
7 Module run	11
7.1 Class Window	11
7.1.1 Methods	11
8 Module test_WGS84DecToDeg	13
8.1 Class WGS84Dec_Test	13
8.1.1 Methods	13
9 Module test_WGS84DegToDec	14
9.1 Class WGS84Deg_Test	14

9.1.1	Methods	14
10	Module test_angleToFunction	15
10.1	Class AngleToFx_Test	15
10.1.1	Methods	15
11	Module test_functionIntersect	16
11.1	Class Intersection	16
11.1.1	Methods	16
12	Module test_mareeCalculator	17
12.1	Class Maree_Test	17
12.1.1	Methods	17
13	Module uiFunctions	18
13.1	Class GpsDialog	18
13.1.1	Methods	18
13.2	Class MareeDialog	18
13.2.1	Methods	18
13.3	Class PositionTheorique	19
13.3.1	Methods	19
13.4	Class CapDialog	19
13.4.1	Methods	20

1 Module WGS84DecToDeg

1.1 Functions

WGS84DecToDeg (<i>Ddec</i>)
Convertis les degrés decimal en degrés sexadecimal
Parameters Ddec: angle en WGS84 decimal (<i>type=float</i>)
Return Value variable WGS84 sexadecimal (<i>type=tuple</i>)
Note: $Ddec = D + M/60 + S/3600$
Since: 0.8
Version: 0.9
Author: Maxime Favier

1.2 Variables

Name	Description
__package__	Value: None

2 Module WGS84DegToDec

2.1 Functions

WGS84DegToDec (<i>D, M, S</i>)
Convertis les degrés sexadecimal en degrés decimal
Parameters
D: WGS84 degrés (<i>type=int</i>)
M: WGS84 minutes (<i>type=int</i>)
S: WGS84 secondes (<i>type=float</i>)
Return Value
variable WGS84 decimal (<i>type=float</i>)
Note: $D_{dec}=D+M/60+S/3600$
Since: 0.9
Version: 0.9
Author: Maxime Favier

2.2 Variables

Name	Description
__package__	Value: None

3 Module *angleToFunction*

3.1 Functions

angleToFunction(*angle*, *point*)

Renvoie l'équation d'une droite à partir d'un point et d'un angle ! le cas ou a = +-inf

Parameters

angle: angle par rapport au nord. (0 - 360)

(*type=float*)

point: point de la droite

(*type=tuple*)

Return Value

a et b de l'équation $f(x)=ax+b$

(*type=tuple*)

Author: Maxime Favier

Since: 0.2

Version: 0.5

3.2 Variables

Name	Description
<code>__package__</code>	Value: None

4 Module functionIntersect

4.1 Functions

functionIntersect (<i>a, b, c, d</i>)
Renvoie le point d'intersection de deux fonctions $f_1(x)=ax+b$ et $f_2(x)=cx+d$
Parameters
a : coef directeur de la droite $f_1(x)$ (<i>type=float</i>)
b : coef de la droite $f_1(x)$ (<i>type=float</i>)
c : coef directeur de la droite $f_2(x)$ (<i>type=float</i>)
d : coef de la droite $f_2(x)$ (<i>type=float</i>)
Return Value
coord du point d'intersection (x,y) (<i>type=tuple</i>)
Author: Maxime Favier
Version: 0.3

4.2 Variables

Name	Description
__package__	Value: None

5 Module map

5.1 Class DrawMap



widget de la carte

Author: Maxime Favier

5.1.1 Methods

<code>__init__(self, im, parentClass)</code> Initialisation du widget pour la carte Parameters im: Chemain d'accès de la carte (<i>type=</i> <i>str</i>) parentClass: classe principale de l'UI (<i>type=</i> <i>Window</i>)
<code>supprimerTraces(self)</code> supprime le tracé des amers
<code>paintEvent(self, event)</code> impressions des formes sur la carte Parameters event: (<i>type=</i> <i>QPaintEvent</i>)
<code>amerCreation(self, event)</code> Creation du trace des amers et determination de la position Parameters event: objet clic de souris (<i>type=</i> <i>QMouseEvent</i>)
<code>gpsDialogManager(self)</code> Hook pour la fenetre de dialogue GPS

getGpsCoordinates(*self*, *ncord*, *wcord*)

Affiche le point GPS sur la carte
Parameters
ncord: Cordonée sexadecimal Nord

(type=tuple)
wcord: Cordonée sexadecimal ouest

(type=tuple)

computeAmerGPSError(*self*)

Calcul l'erreur entre le relevé des amers et la position GPS

mousePressEvent(*self*, *event*)

Hook evenement clic de souris
Parameters
event: objet clic de souris

(type=QMouseEvent)

mareeDialogManager(*self*)

Hook pour la fenetre de dialogue marée

mareeProcessing(*self*, *TMarreeHaute*, *HMarreeHaute*, *TMarreeBasse*, *HMarreeBasse*, *time*)

Manager de calcul des informations de marée
Parameters
TMarreeHaute: heure de maree haute en min

(type=int)
HMarreeHaute: hauteur de l'eau à maree haute

(type=float)
TMarreeBasse: heure de maree basse en min

(type=int)
HMarreeBasse: hauteur de l'eau à maree basse en min

(type=float)
time: temps de la journee

(type=int)

posTheoriqueDialogManager(*self*)

Hook pour la fenetre de dialogue de calcul de la position théorique

posTheoriqueProcessing(*self*, *ncoord*, *wcoord*, *angleCourant*, *vitesseCourant*, *capCompas*, *declinaison*, *deviation*, *deriveVent*, *xMin*)

Manager de calcul des information pour la position théorique

Parameters

ncoord: Cordonée sexadecimal Nord
(*type=tuple*)

wcoord: Cordonée sexadecimal Ouest
(*type=tuple*)

angleCourant: angle du courant
(*type=float*)

vitesseCourant: vitesse du courant m/s
(*type=float*)

capCompas: cap compas (deg)
(*type=float*)

declinaison: declinaison
(*type=float*)

deviation: déviation
(*type=float*)

deriveVent: derive du vent
(*type=float*)

xMin: min de simulation
(*type=int*)

Author: Maxime Favier

capTheoriqueManager(*self*)

Hook pour la fenetre de dialogue pour le calcul du cap

capTheoriqueProcessing(*self*, *ncoord1*, *wcoord1*, *ncoord2*, *wcoord2*)

Manager du calcul de cap

Parameters

ncoord1: coordonnées deg/min/sec nord du pt A
(*type=tuple*)

wcoord1: coordonnées deg/min/sec ouest du pt A
(*type=tuple*)

ncoord2: coordonnées deg/min/sec nord du pt B
(*type=tuple*)

wcoord2: coordonnées deg/min/sec ouest du pt B
(*type=tuple*)

Author: Maxime Favier

5.2 Class Arrow

Classe pour la création de fleches

Author: Maxime Favier

Note: Inspiré de <https://gist.github.com/reusee/2406975>

5.2.1 Methods

```
__init__(self, painter, pos1, pos2)
```

```
paint(self, painter)
```

6 Module *mareeCalculator*

6.1 Functions

mareCalculator(*TMarreeHaute, HMarreeHaute, TMarreeBasse, HMarreeBasse, time*)

Calcule la hauteur de l'eau

Parameters

TMarreeHaute: heure de maree haute en min

(*type=int*)

HMarreeHaute: hauteur de l'eau à maree haute

(*type=float*)

TMarreeBasse: heure de maree basse en min

(*type=int*)

HMarreeBasse: hauteur de l'eau à maree basse en min

(*type=float*)

time: temps de la journee

(*type=int*)

Return Value

hauteur de l'eau

(*type=float*)

Author: Maxime Favier

Since: 0.4

Version: 0.5

6.2 Variables

Name	Description
__package__	Value: None

7 Module run

7.1 Class Window

PyQt5.QtWidgets.QMainWindow —
run.Window

Classe principale du GUI

Author: Maxime Favier

7.1.1 Methods

__init__ (<i>self</i>)
init_ui (<i>self</i>) definitions des propriétés du GUI
draw_toolbar (<i>self</i>) Creation des boutons de la barre de tache
draw_map (<i>self</i>) positionnement de la carte
draw_dock (<i>self</i>) Initialisation du dock
groupePosition (<i>self</i>) Initialisation des widgets du groupe position du bateau - Amers
groupeGPS (<i>self</i>) Initialisation des widgets du groupe GPS
groupeAmersError (<i>self</i>) Initialisation des widgets du groupe erreur amers
groupeMaree (<i>self</i>) Initialisation des widgets du groupe Marée
groupeCap (<i>self</i>) Initialisation des widgets du groupe Cap

updateLabelsAmer(*self*, *pos1*, *pos2*)

mise à jour des labels du groupe position amer

Parameters

pos1: position en deg, min, sec au nord
(*type=*str)

pos2: position en deg, min, sec a l'ouest
(*type=*str)

updateLabelsGPS(*self*, *pos1*, *pos2*)

mise à jour des labels du groupe position GPS

Parameters

pos1: position en deg, min, sec au nord
(*type=*str)

pos2: position en deg, min, sec a l'ouest
(*type=*str)

updateLabelsError(*self*, *pos1*, *pos2*)

mise à jour des labels du groupe position GPS

Parameters

pos1: position en deg, min, sec au nord
(*type=*str)

pos2: position en deg, min, sec a l'ouest
(*type=*str)

updateMarre(*self*, *lbl1*=None, *lbl2*=None)

mise à jour des labels du groupe marée

Parameters

lbl1: contenu du lbl1
(*type=*str)

lbl2: contenu du lbl2
(*type=*str)

updateCap(*self*, *cap*)

mise à jour des labels du groupe cap

Parameters

cap: contenu du lbl
(*type=*str)

8 Module `test_WGS84DecToDeg`

8.1 Class `WGS84Dec_Test`

```
unittest.TestCase └─  
                    test_WGS84DecToDeg.WGS84Dec_Test
```

Tests automatiques pour la fonction `WGS84DecToDeg` qui convertit les degrés WGS84 decimal en sexadécimal

8.1.1 Methods

test_1 (<i>self</i>)
38.8897 = 38 53' 23"

test_2 (<i>self</i>)
-77.0089 = -77 00' 32"

test_3 (<i>self</i>)
47.494128 = 47 29' 39"

test_4 (<i>self</i>)
3.043041 = 3 2' 35"

9 Module `test_WGS84DegToDec`

9.1 Class `WGS84Deg_Test`

```
unittest.TestCase └─ test_WGS84DegToDec.WGS84Deg_Test
```

Tests automatiques pour la fonction `WGS84DegToDec` qui convertit les degrés WGS84 sexadécimal en décimal

9.1.1 Methods

<code>test_1(self)</code>
<hr/>
<code>38.8897 = 38 53' 23"</code>

<code>test_2(self)</code>
<hr/>
<code>-77.0089 = -77 00' 32"</code>

<code>test_3(self)</code>
<hr/>
<code>47.494128 = 47 29' 39"</code>

<code>test_4(self)</code>
<hr/>
<code>3.043041 = 3 2' 35"</code>

10 Module `test_angleToFunction`

10.1 Class `AngleToFx_Test`

```
unittest.TestCase └─  
                    test_angleToFunction.AngleToFx_Test
```

tests automatiques de `angleToFunction` qui renvoie l'équation affine d'une droite à partir de l'angle par rapport au nord et la position d'un point

10.1.1 Methods

test_angle0(<i>self</i>)
teste que quand l'angle est 0, a tend vers l'infini

test_angle180(<i>self</i>)
teste quand l'angle est 180, a tend vers moins l'infini

test_angle45(<i>self</i>)
teste quand l'angle est 45, a=1

test_angle225(<i>self</i>)
teste quand l'angle est 225 au point (1,1), a=1 et b=2

test_angle315(<i>self</i>)
teste quand l'angle est 315 au point (1,0), a=-1 et b=-1

11 Module `test_functionIntersect`

11.1 Class `Intersection`

`unittest.TestCase` — `test_functionIntersect.Intersection`

tests automatiques de la fonction d'intersection qui calcule le point d'intersection avec de deux fonctions affines

11.1.1 Methods

`test_droitesParalleles(self)`

teste que la fonction retourne une erreur lorsque les deux droites sont parralles

`test_intersection00(self)`

teste que la fonction retourne (0,0) comme point d'intersection pour les fonctions $y=x$ et $y=-4x$

`test_intersection12(self)`

teste que la fonction retourne (1,2) comme point d'intersection pour les fonctions $y=-2x+4$ et $y=3x-1$

`test_intersection52(self)`

teste que la fonction retourne (5,-2) comme point d'intersection pour les fonctions $y=-5/7x+11/7$ et $y=3/5x-5$

12 Module `test_mareeCalculator`

12.1 Class `Maree_Test`

`unittest.TestCase` —
`test_mareeCalculator.Maree_Test`

tests automatiques de la fonction qui calcule la hauteur de l'eau à toute heure avec les informations de marée

12.1.1 Methods

`test_maree1(self)`

v. les exercices pour préparer le permis hauturier p17-18 BM 4h57 3.45m PM 11h03 7.05m
la hauteur de l'eau à 7h45 est +-5.03m

`test_maree2(self)`

v. les exercices pour préparer le permis hauturier p19-20 BM 13h18 2.25m PM 19h06 5.85m
la hauteur de l'eau à 16h20 est +-4.18m

`test_maree3(self)`

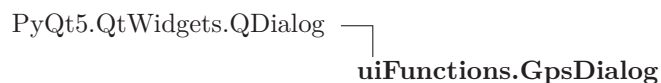
v. les exercices pour préparer le permis hauturier p23-24 PM 18h14 5.70m BM 12h08 1.85m
la hauteur de l'eau à 15h est +- 3.60m

`test_maree4(self)`

v. les exercices pour préparer le permis hauturier p25-26 PM 12h04 7.3m BM 17h52 1.6m la
hauteur de l'eau à 16h10 est +-2.81

13 Module uiFunctions

13.1 Class GpsDialog



Interface de dialogue pour le GPS

Author: Maxime Favier

13.1.1 Methods

init__ (<i>self</i> , <i>parent</i>)
Initialisation des widgets
Parameters
<i>parent</i> : class parente DrawMap (<i>type=DrawMap</i>)

ok (<i>self</i>)
Fonction Valider

13.2 Class MareeDialog



Interface de dialogue pour le calculateur de marée

Author: Maxime Favier

13.2.1 Methods

init__ (<i>self</i> , <i>parent</i> , <i>gparent</i>)
Initialisation des widgets
Parameters
<i>parent</i> : Classe parente DrawMap (<i>type=DrawMap</i>)
<i>gparent</i> : Classe de l'UI (<i>type=Window</i>)

pleineMerGroupe (<i>self</i>)
--

Initialisation des widgets pleine mer

basseMerGroupe (<i>self</i>)

Initialisation des widgets basse mer

ok (<i>self</i>)

Fonction valider

13.3 Class PositionTheorique

PyQt5.QtWidgets.QDialog —
uiFunctions.PositionTheorique

Interface graphique pour le calcul de la position theorique

Author: Maxime Favuer

13.3.1 Methods

__init__ (<i>self</i> , <i>parent</i> , <i>gparent</i>)
--

Initialisation des widgets

Parameters

parent: Classe parente DrawMap
(type=DrawMap)

gparent: Classe de l'UI
(type=Window)

courantGroupe (<i>self</i>)

capGroupe (<i>self</i>)

TempsGroupe (<i>self</i>)

ok (<i>self</i>)

13.4 Class CapDialog

PyQt5.QtWidgets.QDialog —
uiFunctions.CapDialog

Interface de dialogue pour le calcul du cap

Author: Maxime Favier

13.4.1 Methods

<code>__init__</code> (<i>self</i> , <i>parent</i>)
Initialisation des widgets
Parameters
<i>parent</i> : class parente DrawMap (<i>type=DrawMap</i>)

<code>ok</code> (<i>self</i>)
Fonction Valider

Index

- angleToFunction (*module*), 4
 - angleToFunction.angleToFunction (*function*), 4
- functionIntersect (*module*), 5
 - functionIntersect.functionIntersect (*function*), 5
- map (*module*), 6–9
 - map.Arrow (*class*), 8–9
 - map.Arrow.__init__ (*method*), 9
 - map.Arrow.paint (*method*), 9
 - map.DrawMap (*class*), 6–8
 - map.DrawMap.__init__ (*method*), 6
 - map.DrawMap.amerCreation (*method*), 6
 - map.DrawMap.capTheoriqueManager (*method*), 8
 - map.DrawMap.capTheoriqueProcessing (*method*), 8
 - map.DrawMap.computeAmerGPSError (*method*), 7
 - map.DrawMap.getGpsCoordinates (*method*), 6
 - map.DrawMap.gpsDialogManager (*method*), 6
 - map.DrawMap.mareeDialogManager (*method*), 7
 - map.DrawMap.mareeProcessing (*method*), 7
 - map.DrawMap.mousePressEvent (*method*), 7
 - map.DrawMap.paintEvent (*method*), 6
 - map.DrawMap.posTheoriqueDialogManager (*method*), 7
 - map.DrawMap.posTheoriqueProcessing (*method*), 7
 - map.DrawMap.supprimerTraces (*method*), 6
- mareeCalculator (*module*), 10
 - mareeCalculator.marreCalculator (*function*), 10
- run (*module*), 11–12
 - run.Window (*class*), 11–12
 - run.Window.__init__ (*method*), 11
 - run.Window.draw_dock (*method*), 11
 - run.Window.draw_map (*method*), 11
 - run.Window.draw_toolbar (*method*), 11
 - run.Window.groupeAmersError (*method*), 11
 - run.Window.groupeCap (*method*), 11
 - run.Window.groupeGPS (*method*), 11
 - run.Window.groupeMaree (*method*), 11
 - run.Window.groupePosition (*method*), 11
 - run.Window.init_ui (*method*), 11
 - run.Window.updateCap (*method*), 12
 - run.Window.updateLabelsAmer (*method*), 11
 - run.Window.updateLabelsError (*method*), 12
 - run.Window.updateLabelsGPS (*method*), 12
 - run.Window.updateMarre (*method*), 12
- test_angleToFunction (*module*), 15
 - test_angleToFunction.AngleToFx_Test (*class*), 15
 - test_angleToFunction.AngleToFx_Test.test_angle0 (*method*), 15
 - test_angleToFunction.AngleToFx_Test.test_angle180 (*method*), 15
 - test_angleToFunction.AngleToFx_Test.test_angle225 (*method*), 15
 - test_angleToFunction.AngleToFx_Test.test_angle315 (*method*), 15
 - test_angleToFunction.AngleToFx_Test.test_angle45 (*method*), 15
- test_functionIntersect (*module*), 16
 - test_functionIntersect.Intersection (*class*), 16
 - test_functionIntersect.Intersection.test_droitesParalleles (*method*), 16
 - test_functionIntersect.Intersection.test_intersection00 (*method*), 16
 - test_functionIntersect.Intersection.test_intersection12 (*method*), 16
 - test_functionIntersect.Intersection.test_intersection52 (*method*), 16
- test_mareeCalculator (*module*), 17
 - test_mareeCalculator.Maree_Test (*class*), 17
 - test_mareeCalculator.Maree_Test.test_maree1 (*method*), 17
 - test_mareeCalculator.Maree_Test.test_maree2 (*method*), 17
 - test_mareeCalculator.Maree_Test.test_maree3 (*method*), 17
 - test_mareeCalculator.Maree_Test.test_maree4 (*method*), 17
- test_WGS84DecToDeg (*module*), 13
 - test_WGS84DecToDeg.WGS84Dec_Test (*class*), 13
 - test_WGS84DecToDeg.WGS84Dec_Test.test_1 (*method*), 13
 - test_WGS84DecToDeg.WGS84Dec_Test.test_2 (*method*), 13
 - test_WGS84DecToDeg.WGS84Dec_Test.test_3 (*method*), 13
 - test_WGS84DecToDeg.WGS84Dec_Test.test_4 (*method*), 13
- test_WGS84DegToDec (*module*), 14
 - test_WGS84DegToDec.WGS84Deg_Test (*class*), 14

- test_WGS84DegToDec.WGS84Deg_Test.test_1
(method), 14
- test_WGS84DegToDec.WGS84Deg_Test.test_2
(method), 14
- test_WGS84DegToDec.WGS84Deg_Test.test_3
(method), 14
- test_WGS84DegToDec.WGS84Deg_Test.test_4
(method), 14
- uiFunctions (module), 18–20
 - uiFunctions.CapDialog (class), 19–20
 - uiFunctions.CapDialog.__init__ (method), 20
 - uiFunctions.CapDialog.ok (method), 20
 - uiFunctions.GpsDialog (class), 18
 - uiFunctions.GpsDialog.__init__ (method), 18
 - uiFunctions.GpsDialog.ok (method), 18
 - uiFunctions.MareeDialog (class), 18–19
 - uiFunctions.MareeDialog.__init__ (method),
18
 - uiFunctions.MareeDialog.basseMerGroupe (method),
19
 - uiFunctions.MareeDialog.ok (method), 19
 - uiFunctions.MareeDialog.pleineMerGroupe (method),
18
 - uiFunctions.PositionTheorique (class), 19
 - uiFunctions.PositionTheorique.__init__ (method),
19
 - uiFunctions.PositionTheorique.capGroupe (method),
19
 - uiFunctions.PositionTheorique.courantGroupe
(method), 19
 - uiFunctions.PositionTheorique.ok (method), 19
 - uiFunctions.PositionTheorique.TempsGroupe (method),
19
- WGS84DecToDeg (module), 2
 - WGS84DecToDeg.WGS84DecToDeg (function),
2
- WGS84DegToDec (module), 3
 - WGS84DegToDec.WGS84DegToDec (function),
3