

**dreamtools**

*Release 1.0.3*

**dreamgeeker**

janv. 07, 2021



<b>1</b>	<b>Dreamtools</b>	<b>1</b>
1.1	Installation . . . . .	1
1.2	Configuration . . . . .	1
1.3	Crédits . . . . .	1
<b>2</b>	<b>Gestion configuration</b>	<b>3</b>
2.1	Repertoires par défaut . . . . .	3
2.2	Class CFBases . . . . .	3
2.3	Class CReponder . . . . .	5
2.4	Class CTracker . . . . .	5
<b>3</b>	<b>Gestions des logs</b>	<b>7</b>
<b>4</b>	<b>Librairie de fonctions</b>	<b>9</b>
4.1	Module de fonctions basiques . . . . .	9
4.2	Module complémentaire . . . . .	15
<b>5</b>	<b>Module de validation</b>	<b>17</b>
<b>6</b>	<b>Gestion de mailing</b>	<b>19</b>
6.1	Pré-Requis . . . . .	19
6.2	Class CMailer . . . . .	19
<b>7</b>	<b>Traitement d'image</b>	<b>21</b>
<b>8</b>	<b>Module de cryptage</b>	<b>23</b>
<b>9</b>	<b>Module de Gestion des date</b>	<b>25</b>
9.1	Constantes globales . . . . .	25
9.2	Fonctions . . . . .	25
	<b>Index des modules Python</b>	<b>31</b>
	<b>Index</b>	<b>33</b>

---



# Dreamtools

Dreamtools est un outils d'aide au développement contenant une liste de fonction d'utilisation basique

## 1.1 Installation

```
$ pip install dreamtools
$ tools-installer
*****
** Création architecture
** -----
** Répertoire logs
**   >> Répertoire créé : C:\Users\Ohanna\Geekspace\testpack\logs
** Répertoire configuration
**   >> Répertoire créé : C:\Users\Ohanna\Geekspace\testpack\cfg
** =====
```

Le répertoire de configuration “cfg” sera créé à la racine du projet.

**Avertissement** Initiliser les données de l’application

## 1.2 Configuration

```
import dreamtools

app_name = "AMON_APP"    #nom de votre application
dreamtools.config(app_name, mode='DEBUG') # par défaut mode ='PROD'
```

**Avertissement** Le paquet comprend un module de cryptage non supporté par Winddows

## 1.3 Crédits

Conçut par Dreamgeerker



---

## Gestion configuration

---

Gestion fichiers de configurations (YAML)

pathfile : dreamtools/cfgmng.py

### 2.1 Repertoires par défaut

---

#### Note

- PROJECT\_DIR/cfg/PROJECT\_DIR/cfg/.log.yml : Fichier de configuration des logs
  - PROJECT\_DIR/cfg/.app.yml : Fichier de configuration de l'application
  - PROJECT\_DIR/cfg/categorie.yml : Fichier de liste définie par un code et un libelle
  - PROJECT\_DIR/cfg/mailling.yml : Fichier de mails préparés
  - PROJECT\_DIR/cfg/validators.yml : Fichier de validation(cf CERBERUS)
  - PROJECT\_DIR/cfg/normalizor.yml : Fichier de normalization(cf CERBERUS)
- 

### 2.2 Class CFBases

#### class CFBases

Cette class permet de gere des fichiers de configuration disponibles dans le repertoire <PROJECT\_DIR>/cfg

**static app\_cfg** ( *code=None* )

Parametres application

**Paramètres code** (*str*) – clé a retourner (filtre)

**Renvoie** Configuration

**static categorie\_lib** ( *code=None* )

Liste de definition

**Paramètres code** (*str*) – référence du de la liste

**Renvoie** liste(s) de categories

Type renvoyé dict

**static loadingbyref** ( *filename*, \*args, \*\*kwargs )

Récupération des parametres de configuration du fichier <filepath> section <r>

Paramètres **filename** (*str*) – Fichier de configuration

**static logs\_cfg** ( )

Configuration des logs

Exemple

```
>>> import logging.config as log_config
>>> import logging
>>> log_config.dictConfig(CFGBases.logs_cfg())
>>> tracker = logging.getLogger('PROD|TEST')
>>> tracker.info("Exemple dun message d'information")
```

**static mailing\_lib** ( *code* )

Mail préparé

Paramètres **code** (*str*) – référence du mail à envoyer

Renvoie mail

**static normalizer** ( )

Parametres de normalisation de formulaire :return: parametres de normaisation :rtype: dict

**static validator** ( )

Parametres de validation de formulaire

**class CFGEEngine**

cfg engine

**static initial** ( *baz\_dir*='cfg' )

Paramètres **baz\_dir** (*str*) –

Renvoie

**static loading** ( *p*, *ref*=None, *m*='r' )

Récupération des parametres de configuration du fichier <p> section <r>

Paramètres • **p** (*str*) – Fichier de configuration

• **ref** (*str*) – référence parametres à récupérer, optionnel

• **m** (*str*) – bytes par default

Renvoie configuration | None

**static save\_cfg** ( *d*, *f*, *m*='w' )

Paramètres • **list(str)** **d** (*dict* (*str*), ) – données à enregistrer

• **f** (*str*) – nom du fichier

• **m** (*str*) – default (write): mode « w | a », optional

Renvoie



## 2.3 Class CReponder

## 2.4 Class CTracker

Module de gestion des logs :

- Récupération des logs, traitement
- Execution sécurisé



---

## Gestions des logs

---

- Configuration des logs
- Traitement des erreurs et des exceptions

pathfile : dreamtools/logmng.py

**exception CError** ( *message, status, title='ERRCustom'* )

Gestion des erreurs et traitement des exceptions personnalisées



---

## Librairie de fonctions

---

### 4.1 Module de fonctions basiques

Liste de fonctions utiles

pathfile : dreamtools/tools

#### 4.1.1 Constantes globales

---

##### Note

- `RGX_ACCENTS` = “àâãäèéëîïôöððùüÿñç”
- `RGX_EMAIL` = Expression reguliere email
- `RGX_PUNCT` = Caractere speciaux autorisé pour mot de passe
- `RGX_PWD` = Expression régulière pour un mot de passe de 8 à 12 avec un car.Special/une Majuscule/Une minuscule
- `RGX_PHONE` = Expression réguliere remative à un numéro de téléphon
- `RGX_URL` = expression reguliere pour URL
- `PROJECT_DIR` : Repertoire du projet
- `APP_NAME` : Nom de l’application
- `APP_DIR` : `PROJECT_DIR/APP_NAME`
- `TMP_DIR` : `PROJECT_DIR/tmp`

---

**Avertissement** Il faut configurer l’application afin d’avoir accès au variable `PROJECT_DIR`, `APP_NAME`, `TMP_DIR`

```
>>> from dreamtools import config
>>> from dreamtools import tools
>>> config.CConfig('monapp')
>>> print (tools.APP_DIR)
'../PROJECT/mon_app'
```

**add\_list** ( *v*, *ll* )

Ajout d'un item dans une liste avec gestion des doublons

- Paramètres**
- **v** (*str*) – valeur à ajouter
  - **ll** (*list*) – liste

**addhex** ( *h*, *v* )

Additionne une valeur hexadecimal

- Paramètres**
- **h** (*str*) – valeur hexadecimal
  - **v** (*int*) – valeur entière à ajouter

**Renvoie** valeur additionné en hexedécimal

**Exemple**

```
>>> hx = '0x129'
>>> addhex(hx, 2)
0x12b
```

**aleatoire** ( *end*, *s=1* )

Génération d'un nombre aléatoire entre [1-end] => end caractère

- Paramètres**
- **end** (*int*) – valeur maximal (paut indiquer la taille si s=1)
  - **s** – valeur de départ, default to 1

**Renvoie** Un chiffre aléatoire

**Exemple**

```
>>> aleatoire (5)
1 : Renvoie un chiffre entre 1 et 5
>>> aleatoire (5,3)
1 : Renvoie un chiffre entre 3 et 5
>>> 4
```

**check\_password** ( *s* )

Vérifie que la syntaxe d'une chaine répond au critère d'un mot de passe

- Conditions**
- Une majuscule
  - Une minuscule
  - Un chiffre
  - Un carectère spécial (@#!?\${&\_- autorisé )

**Paramètres** **s** (*str*) – chaine à vérifier

**Return bool** True si la chaine est valide

**clean\_allspace** ( *ch*, *very\_all=True* )

Nettoyage de tous les espace et carateres vides

- Paramètres**
- **ch** (*str*) – Chaine à nettoyer
  - **very\_all** (*bool*) – caractère vide aussi, True (False = Espaces uniquement)

**Exemple**

```
>>> chaine = 'Se réveiller au matin de sa destiné !'
>>> clean_allspace (chaine)
```

```
'Seréveilleraumatindesadestiné!'
```

**clean\_coma** ( *ch*, *w\_punk=False* )

Supprime les accents/caractères spéciaux du texte source en respectant la casse

**Paramètres**

- **ch** – Chaîne de caractere à « nettoyer »
- **w\_punk** – indique si la punctuation est à nettoyer ou pas (suppression)

**Exemple**

```
>>> s = 'Se réveiller au matin de sa destiné !!'
>>> clean_coma (s)
'Se seveiller au matin (ou pas) de sa destine !!!'
>>> clean_coma (s, True)
'Se reveiller au matin ou pas de sa destine'
```

**clean\_dir** ( *directory*, *pattern='\*'* )

Supprime tous les éléments d'un repertoire

**Paramètres**

- **directory** (*str*) – chemin du repertoire
- **pattern** (*string*) – patter des fichier à supprimer (filtre)

**Return int** nombre de fichier supprimer

**clean\_master** ( *ch* )

Supprime les accents, caractères spéciaux et espace du texte source

**Paramètres** **ch** (*str*) – Chaîne de caractere à « nettoyer »

**Return str** chaîne sans accents:car. spéciaux ni espace en minuscule

**Exemple**

```
>>> s = 'Se réveiller au matin (ou pas) de sa destiné !'
>>> clean_master (s)
'sereveilleraumatinoupasdesadestine'
```

**clean\_space** ( *ch* )

Nettoyage des espaces « superflus »

- Espaces à gauche et à droite supprimés
- Répétition d'espace réduit

**Exemple**

```
>>> chaine = 'Se réveiller au matin de sa destiné ! !'
>>> clean_space (chaine)
'Se réveiller au matin de sa destiné ! !'
```

**code\_maker** ( *i\_size=4* )

Génération d'une chaîne aléatoire composé de lettre et de chiffres

**Paramètres** **i\_size** (*int*) – taille du code  
**Rtype str**

**comphex** ( *hx\_a*, *hx\_b* )

Compare deux valeurs hexadécimales

**Paramètres**

- **hx\_a** (*str*) –

- **hx\_b** (*str*) –

**Return int**

- 0 : *hx\_a* == *hx\_b*
- 1 : *hx\_a* > *hx\_b*
- -1 : *hx\_a* < *hx\_b*

**dicFindKey** (*value*, *dic*)

Recherche une clé d'un dictionnaire à partir de sa valeur

**dictlist** (*k*, *v*, *d*)

Ajout d'un valeur dans une liste d'un dictionnaire

**Paramètres**

- **k** (*str*) – clé dictionnaire
- **v** – valeur à ajouter
- **list []** **d** (*dict* [*str*],) – dictionnaire

**Exemple**

```
>>> dictionnaire= {}
>>> dictlist('printemps', 'mar', dictionnaire)
dictionnaire{'printemps', ['mars']}
>>> dictlist('printemps', 'avril', dictionnaire)
dictionnaire{'printemps', ['mars', 'avril']}
>>> dictlist('printemps', 'mars', dictionnaire)
dictionnaire{'printemps', ['mars', 'avril']}
```

**dirparent** (*path*)

Renvoie du repertoire parent

**Paramètres** **path** (*str*) – repertoire

**Type renvoyé** str

**dirparser** (*directory*, *pattern*='\*')

Récupération des fichiers d'un répertoire

**Paramètres**

- **directory** (*str*) – repertoire
- **pattern** (*str*) – "\*" pour tous type de fichier par défaut

**Exemple**

```
>>> directory = 'C:\Users\public\Documents'
>>> pattern='*.txt'
>>> for filename, path_file in dirparser(directory, pattern):
...     print(path_file)
'C:\Users\public\Documents\fichier.txt'
'C:\Users\public\Documents\autre_fichier.txt'
```

**dirproject** ( )

Répertoire d'exécution

**dirprojet** ( )

Répertoire pour le fichier en cours

**Type renvoyé** str



**file\_exists** (*fp*)

Vérifie l'existence d'un fichier

**Paramètres** **fp** (*str*) – filepath**Rtype** bool**file\_ext** (*ps\_file*)

Retourne l'extension d'un fichier

**Paramètres** **ps\_file** –**Renvoie** Extension de fichier**inttohex** (*v*)

Conversion d'une valeur en hexadécimal

**Paramètres** **v** (*int*) – nombre à convertir**Renvoie** valeur en hexadécimal**Type renvoyé** str**makedirs** (*path*)

Création du répertoire données

**Paramètres** **path** – chemin du répertoire à créer**Rtype** bool**path\_build** (*directory, ps\_complement*)

Construction d'un pathfile

**Paramètres** • **directory** (*str*) – repertoire• **ps\_complement** (*str*) – complement permettant de generer le chemin**Type renvoyé** str**Exemple**

```
>>> path = 'c:\Users\public\directory'
>>> path_build(path, '..\other_dir')
'c:\Users\public\other_dir'
```

**plain\_hex** (*hx, s=3*)

Complète un chiffre hexadécimal en préfixant une valeur de zéro

**Paramètres** • **hx** (*str*) – valeur hexadécimal• **s** (*int*) – longueur chaine attendu**Type renvoyé** str:**Exemples**

```
>>> hx = '0x129'
>>> plain_hex(hx, 5)
0x00129
```

**plain\_zero** (*v, s*)

Complete une valeur chaine de zéro

**Paramètres** • **v** – valeur à completer

- **s** – taille chaine attendu préfixé de zerom

**Exemple**

```
>>> d = 5
>>> plain_zero(d, 3)
'005'
```

**pop\_dic** (*l\_ids, dic*)

Suppression d'une liste d'éléments d'un dictionnaire

:param list[str] *l\_ids* : liste de clé à supprimer :param dict[str:object] *dic*: dictionnaire à nettoyer

**print\_err** (*\*args, \*\*kwargs*)

Ecriture sur le flux erreur de la console

**Paramètres**

- **args** – arguments 1
- **kwargs** – arguemnts2

**pwd\_maker** (*i\_size=8*)

Génération d'un password respectant les regles de password

**Conditions**

- Une majuscule
- Une minuscule
- Un chiffre
- Un carectère spécial (@#!?\${&\_- autorisé )

**Paramètres** *i\_size* (*int*) – Nombre de caracteres de la chaine

**Renvoie** Mot de passe

**remove\_file** (*p*)

Suppression d'un fichier si existant

**Paramètres** *p* (*str*) – chemin complet du fichier à supprimer

**str\_dic** (*chaine*)

Conversion d'une chaine en dictionnaire

**Paramètres** *chaine* (*str*) –

**Type renvoyé** dic

**Exemple**

```
>>> s_dic = '{"key':value}"
>>> str_dic(s_dic)
{'key': 'value'}
```

**string\_me** (*v*)

Conversion d'une valeur en chaine

**Paramètres** *v* – valeur à convertir

**Type renvoyé** str, None en cas d'erreur

## 4.2 Module complémentaire

pathfile : dreamtools/features.py

**test\_http\_link** ( *url* )

Vérifie une url et renvoie l'url valide

**Paramètres** **url** – url à évaluer

**Rtype** str

**url\_join** ( *domaine*, *page* )

Generation d'une url



---

## Module de validation

---

Validation de formulaire.

**Note** Les schémas de validation sont sauvegarder dans le dossier de configuration validators Les schéma de normalization dans normalizator

---

**class Validata** ( *scheme*, \*args, \*\*kwargs )

Validators

**static check\_post\_data** ( *data*, *form\_ref* )

verification donnees formulaire reçu

**Paramètres**

- **data** (*dict*) – formulaire de données
- **form\_ref** (*str*) – reference validateur

**Renvoie**    formulaire traité

**normalisation** ( *document* )

**Paramètres**

- **document** –
- **args** –
- **kwargs** –

**Renvoie**

**validation** ( *document*, \*args, \*\*kwargs )

**Paramètres**

- **document** –
- **args** –
- **kwargs** –

**Renvoie**



---

## Gestion de mailing

---

Module de Gestion de mail préparés

pathfile : dreamtools/mailbot.py

### 6.1 Pré-Requis

**Avertissement** Indiquer les parametres smtp dans le fichiers de configuration <PROJECT\_NAME>/cfg/.app.yml

```
smtp:
  h: smtp-host_adresse
  po: port_smtp
  m: mail_authen
  pw: password_auth
  h_s : name_sender <email>
```

**Avertissement** Les mails sont à définir dans le fichier <PROJECT\_NAME>/cfg/mailling.yml au format suivant

```
footer:
  html: <Pied de mail unique pour tous les mails (signature, rgpd...)>
  text: <Pied de mail unique pour tous les mails (signature, rgpd...)>
code_mail:
  html: <ici mail au format HTML>
  text : <Le mail au format texte>
  objt : <Objet du mail>
```

### 6.2 Class CMailer

class CMailer

**static presend** ( email, code, name='', \*\*data\_field )

Preparation pour envoi d'un message mail

**Paramètres** • **email** (str) – email destinataire

- **code** (*str*) – référence du mail à chargé
- **name** (*str*) – nom du destinataire
- **data\_field** (*dict*) – liste de données relatif à des champs définis dans le mails



---

## Traitement d'image

---

**TYPE\_IMG\_GIF** = 'GIF'

Class permettant le traitement d'une image png convertit en jpg avec prise en charge de la transparence (fond blanc)

pathfile : dreamtools/pyimaging

**treat\_dir** ( *s* )

Redimensionne toutes les images contenu dans un répertoire donné + thumb :param s:

**treat\_img** ( *s,f* )

Enregistre une image, la reformat + thumb :param s: :param f:

**class CImagine** ( *src, dest, with\_ext=False* )

**protected** ( *artist* )

Ajoute un nom d'artist et le copyright d'une image

**resize** ( *s=None, mn=None, mx=None* )

Redimensionnement de l'image au format jpg

- Paramètres**
- **int**) *s* (*tuple* (*int*,) ) – si indiqué, (w,h) de redimensionnement
  - **mn, optional** (*int*) – taille maximum (carré), default 250
  - **mx, optional** (*int*) – taille maximum de l'image, default 250

**Renvoie**

**thumbed** ( *s=None* )

Thumb Image

**Paramètres int**] *s* (*tuple* [*int*,) ) – taille image, defaul (200, 200)

**property white\_background**

Ajout d'un fond transparent :return:



---

## Module de cryptage

---

pathfile : dreamtools/tools

---

**Important** Module utilisable uniquement sous Linux. Pas de prise en charge du module crypt par Windows

---

**Note** Une clé privée et un grain de sel doivent être défini dans le fichier « de paramètre d'application »

- **PROJECT\_DIR/cfg/app.yml**
- SECRET\_KEY
- SALT\_EXT

*Méthode* : sha-512\*

---

**Exemple**

```
>>> from krypt import CKrypting
>>> pwd = CKrypting.encrypt("Mon mon de passe en clair")
>>> if CKrypting.compare(pwd, 'Mon mot de passe en claire'):
>>>     print ('Saisie invalide')
>>> else:
>>>     print ('Bienvenue')
```



---

## Module de Gestion des date

---

Liste de fonction pour utilisation des dates pathfile : dreamtools/tools

### 9.1 Constantes globales

Liste des jours de la semaine I\_MON, I\_TUES, I\_WED, I\_THU, I\_FRI, I\_SAT, I\_SUN = 1, 2, 3, 4, 5, 6, 7

### 9.2 Fonctions

**date\_add\_workday** ( *dte*, *nb* )

Ajoute un nombre de jours ouvrés donnés à une date

**Paramètres**

- **dte** (*datetime*) – date de référence
- **nb** (*int*) – nombre de jour à additionner (valeur négative/positive)

**Renvoie** date de depart + nombre de jours

**date\_dayed** ( *dte=None*, *b=True* )

Positionne la date indiquée à minuit au matin ou au soir

**Paramètres**

- **dte** (*datetime*) – Date
- **b** (*bool*) – Date debut de jour (00:00:00.000) ou date de fin de journee date du jour + 1 (minuit) soit lendemin à 00

**Example**

```
>>> date_dayed()
datetime.datetime(2020, 12, 19, 0, 0, datetime.datetime(2020, 12, 19, 0, 0))
```

**date\_rss** ( *dte=None* )

Dtate au format RSS

**dateadd** ( *dte*, *nb*, *fm='d'* )

Ajoute un nombre de jours données à une date

**Paramètres**

**Type renvoyé** datetime

**datepaques** ( *y* )

Dates Pâques d'une année donnée

- Lundi de paque : lundi suivant le dimanche de paque (La Pâque)
- Jeudi de l'ascension : 3 jour après paques
- pentecote : 49 jours après le lundi de paques

**Paramètres**    **y** (*int*) – année de référence

**Type renvoyé** list[date]

**datestr** ( *dte=None, fm='%Y-%m-%dT%H:%M:%S'* )

Convertit une date en chaine selon un format donnée

**Paramètres**        • **dte** (*datetime*) – date à convertir date du jour par défaut  
                      • **fm** (*str*) – format désirée, defaults to “%d/%m/%Y”

**Renvoie**            Renvoie un chaine correspondant au format date passé en parametre

**Type renvoyé** str

**Exemple**

```
>>> d = maintenant ()
>>> datestr (d, '%d.%m.%Y')
02.06.2019
```

**datetime\_from\_local\_to\_utc** ( *utc\_datetime* )

Convertie une date et heure local en heure utc

**Paramètres** **utc\_datetime** (*datetime*) – datetime local

**Renvoie**    datetime utc

**datetime\_from\_utc\_to\_local** ( *utc\_datetime* )

Convertie la date et heure donné (utc) en date local

**Paramètres** **utc\_datetime** – datetime utc

**Renvoie**    date locale

**day\_in\_hour** ( *dy* )

Conversion d'un nombre de jours en heure

**Paramètres**    **dy** (*int*) – nombre de jours

**Type renvoyé** int

**day\_in\_sec** ( *dy, ml=False* )

Conversion d'un nombre de jours en secondes ou milisecondes

**Paramètres**        • **dy** (*int*) – nombre de jours  
                      • **ml** (*bool*) – en millisecondes si True sinon en secondes, dafault False

**Renvoie**    (milli) secondes

**dtecompare** ( *dtea, dteb* )

Compare si dateb es supérieur à la datea

**Paramètres**        • **dtea** (*datetime*) – date à comparer

- **dteb** (*datetime*) – date à comparer

**Retur** 0 si les dates sont égale, -1 si la dateb est antérieur à datea 1 le c&s inverse

**Type renvoyé** int

**dtediff** ( *dtea, dteb* )

Calcul du nombre de jours entre deux dates

- Paramètres**
- **dtea** (*datetime*) – date à comparer
  - **dteb** (*datetime*) – date à comparer

**Type renvoyé** int

**dtets** ( *dte=None* )

Conversion date - timestamp

**Paramètres** **dte** (*date*) – date à convertir

**Return int** date en milliseconde (sans les ms)

**fullmonth** ( *dte* )

Renvoie la date du jour au format MOIS YYYY

**Paramètres** **dte** (*datetime*) –

**Type renvoyé** str

**get\_date** ( *p\_year, p\_month, p\_day* )

Generation d'une date a partir des valeur numerique

- Paramètres**
- **p\_year** (*int*) – année
  - **p\_month** (*int*) – mois
  - **p\_day** (*int*) – date jour

**Type renvoyé** datetime

**get\_time** ( *dte* )

Renvoie d'une date au format time :param datetime dte: :rtype: time

**get\_weeks\_num** ( *dte=None* )

Renvoie le numéro de la date indiqué (now par deafut)

**is\_workday** ( *dte* )

Determine si la date est un jour ouvré ou vaqué (week-end / fériés)

**Paramètres** **dte** – date à évaluer

**Renvoie** renvoie le statut jour ouvré (true=ouvré)

**Rtype** bool

**isotodate** ( *dte* )

Conversion str\_iso - date

Format ISO : YYYY-MM-DDTHH:MN :param dte:

**Renvoie**

**jours\_feries** ( *y=None* )

Jour fériés pour une date donnée

**Paramètres** *y* (*int*) – Année de référence (optionnel), default : année en cours**Renvoie** un tableau de date de jours fériés**Exemple**

```
>>> jours_feries () #Jours fériés année en cours
>>> jours_feries (2018) # jours fériés année 2018
```

**maintenant** ( *utc=False, fm=None, tz=None* )

Date et heure de l'instant (Now)

**Paramètres**

- **utc** (*bool*) – Si True renvoie de l'heure UTC (GMT) ou l'heure local
- **fm** (*str*) – format date : fm == FRM\_ISO (iso) | FRM\_TIMESTAMP (ts)
- **tz** (*timezone*) – Timezone, None by def

**Type renvoyé** datetime | string**Exemple**

```
>>> maintenant ()
datetime.datetime (2019, 06, 02, 17, 30, 43, 248622)
>>> maintenant (True)
'2019-06-02T17:30:43.248622'
```

**set\_timezone** ( *dt, tz=<UTC>* )

Applique la timezone indiquée à la date passée en parametre

**Paramètres**

- **dt** (*datetime*) – date
- **tz** (*timezone*) – timezone

**strdate** ( *dte, fm='%d-%m-%Y %H:%M:%S'* )

Conversion string - date

**Paramètres**

- **dte** (*str*) – date
- **fm** (*str*) – formayt, default “%d-%m-%Y %H:%M:%S”: patterne, optional

**Renvoie** Renvoie la date convertit ou None en cas d'invalidité (date non conform)**Type renvoyé** datetime**Exemple**

```
>>> s = '24-02-1976 16:45'
>>> strdate (s, '%d-%m-%Y %H:%m')
datetime.datetime (1976, 02, 24, 16, 45)
```

**timeadd** ( *dte, nb* )

Ajoute un nombre d'heure données à une date

**Paramètres**

- **dte** (*date*) – date de départ
- **nb** (*int*) – nombre de jour à additionner (valeur négative/positive)

**Renvoie** date de depat + nombre de jours**today** ( *fm='%d/%m/%Y'* )

Renvoie la date du jour



**Paramètres**    **fm** (*str*) – Format de la date attendu

**Type renvoyé** str

**Exemple**

```
>>> today ()
'02/06/2019'
>>> today ('%d.%m.%Y')
'02.06.2019'
```

**tsdate** (*ts*)

Conversion timestamp - date

**Paramètres** **ts** – temps en milliseconde depuis 1970

**Renvoie**    date

**tstring** (*ts*, *fm*='%Y.%m.%d-%H:%M (%a)')

Conversion timestamp - chaine(str)

**Paramètres**

- **ts** (*int*) – timestamp
- **fm** (*str*) – format attendu

**Type renvoyé** str

**utcnow\_iso** ( )

Date et heure actuelle utc au format iso :return: date utc

**utcnow\_ts** ( )

Date et heure actuelle utc au format timestamp :return: timestamp

- [Index](#)
- [Index du module](#)
- [Page de recherche](#)



## d

### dreamtools

- dreamtools.cfgmng, 3
- dreamtools.dtemng, 23
- dreamtools.features, 15
- dreamtools.imagine, 21
- dreamtools.logmng, 7
- dreamtools.mailbot, 19
- dreamtools.tools, 9
- dreamtools.validata, 17



**A**

`add_list()` (dans le module `dreamtools.tools`), 10  
`addhex()` (dans le module `dreamtools.tools`), 10  
`aleatoire()` (dans le module `dreamtools.tools`), 10  
`app_cfg()` (méthode statique `CFGBases`), 3

**C**

`categorie_lib()` (méthode statique `CFGBases`), 3  
`CError`, 7  
`CFGBases` (classe dans `dreamtools.cfgmng`), 3  
`CFGEngine` (classe dans `dreamtools.cfgmng`), 4  
`check_password()` (dans le module `dreamtools.-tools`), 10  
`check_post_data()` (méthode statique `Validata`), 17  
`CImagine` (classe dans `dreamtools.imagine`), 21  
`clean_allspace()` (dans le module `dreamtools.-tools`), 10  
`clean_coma()` (dans le module `dreamtools.tools`), 11  
`clean_dir()` (dans le module `dreamtools.tools`), 11  
`clean_master()` (dans le module `dreamtools.-tools`), 11  
`clean_space()` (dans le module `dreamtools.tools`), 11  
`CMailer` (classe dans `dreamtools.mailbot`), 19  
`code_maker()` (dans le module `dreamtools.tools`), 11  
`comphex()` (dans le module `dreamtools.tools`), 11

**D**

`date_add_workday()` (dans le module `dreamtools.dtemng`), 25  
`date_dayed()` (dans le module `dreamtools.dtemng`), 25  
`date_rss()` (dans le module `dreamtools.dtemng`), 25  
`dateadd()` (dans le module `dreamtools.dtemng`),

25

`datepaques()` (dans le module `dreamtools.dtemng`), 26  
`datestr()` (dans le module `dreamtools.dtemng`), 26  
`datetime_from_local_to_utc()` (dans le module `dreamtools.dtemng`), 26  
`datetime_from_utc_to_local()` (dans le module `dreamtools.dtemng`), 26  
`day_in_hour()` (dans le module `dreamtools.dtemng`), 26  
`day_in_sec()` (dans le module `dreamtools.dtemng`), 26  
`dicFindKey()` (dans le module `dreamtools.tools`), 12  
`dictlist()` (dans le module `dreamtools.tools`), 12  
`dirparent()` (dans le module `dreamtools.tools`), 12  
`dirparser()` (dans le module `dreamtools.tools`), 12  
`dirproject()` (dans le module `dreamtools.tools`), 12  
`dirprojet()` (dans le module `dreamtools.tools`), 12  
`dreamtools.cfgmng`  
  module, 3  
`dreamtools.dtemng`  
  module, 23  
`dreamtools.features`  
  module, 15  
`dreamtools.imagine`  
  module, 21  
`dreamtools.logmng`  
  module, 7  
`dreamtools.mailbot`  
  module, 19  
`dreamtools.tools`  
  module, 9  
`dreamtools.validata`  
  module, 17  
`dtecompare()` (dans le module `dreamtools.dtemng`), 26  
`dtediff()` (dans le module `dreamtools.dtemng`), 27  
`dtets()` (dans le module `dreamtools.dtemng`), 27

## F

file\_exists() (dans le module dreamtools.tools), 13  
file\_ext() (dans le module dreamtools.tools), 13  
fullmonth() (dans le module dreamtools.dtemng), 27

## G

get\_date() (dans le module dreamtools.dtemng), 27  
get\_time() (dans le module dreamtools.dtemng), 27  
get\_weeks\_num() (dans le module dreamtools.dtemng), 27

## I

initial() (méthode statique CFGEngine), 4  
inttohex() (dans le module dreamtools.tools), 13  
is\_workday() (dans le module dreamtools.dtemng), 27  
isotodate() (dans le module dreamtools.dtemng), 27

## J

jours\_feries() (dans le module dreamtools.dtemng), 28

## L

loading() (méthode statique CFGEngine), 4  
loadingbyref() (méthode statique CFGBases), 4  
logs\_cfg() (méthode statique CFGBases), 4

## M

mailing\_lib() (méthode statique CFGBases), 4  
maintenant() (dans le module dreamtools.dtemng), 28  
makedirs() (dans le module dreamtools.tools), 13  
module  
    dreamtools.cfgmng, 3  
    dreamtools.dtemng, 23  
    dreamtools.features, 15  
    dreamtools.imagine, 21  
    dreamtools.logmng, 7  
    dreamtools.mailbot, 19  
    dreamtools.tools, 9  
    dreamtools.validata, 17

## N

normalisation() (méthode Validata), 17  
normalizer() (méthode statique CFGBases), 4

## P

path\_build() (dans le module dreamtools.tools), 13  
plain\_hex() (dans le module dreamtools.tools), 13  
plain\_zero() (dans le module dreamtools.tools), 13  
pop\_dic() (dans le module dreamtools.tools), 14  
presend() (méthode statique CMailer), 19  
print\_err() (dans le module dreamtools.tools), 14  
protected() (méthode CImagine), 21  
pwd\_maker() (dans le module dreamtools.tools), 14

## R

remove\_file() (dans le module dreamtools.tools), 14  
resize() (méthode CImagine), 21

## S

save\_cfg() (méthode statique CFGEngine), 4  
set\_timezone() (dans le module dreamtools.dtemng), 28  
str\_dic() (dans le module dreamtools.tools), 14  
strdate() (dans le module dreamtools.dtemng), 28  
string\_me() (dans le module dreamtools.tools), 14

## T

test\_http\_link() (dans le module dreamtools.features), 15  
thumbed() (méthode CImagine), 21  
timeadd() (dans le module dreamtools.dtemng), 28  
today() (dans le module dreamtools.dtemng), 28  
treat\_dir() (dans le module dreamtools.imagine), 21  
treat\_img() (dans le module dreamtools.imagine), 21  
tsdate() (dans le module dreamtools.dtemng), 29  
tstring() (dans le module dreamtools.dtemng), 29  
TYPE\_IMG\_GIF (dans le module dreamtools.imagine), 21

## U

url\_join() (dans le module dreamtools.features), 15  
utcnw\_iso() (dans le module dreamtools.dtemng), 29  
utcnw\_ts() (dans le module dreamtools.dtemng), 29

## V

Validata (classe dans dreamtools.validata), [17](#)

validation() (méthode Validata), [17](#)

validator() (méthode statique CFGBases), [4](#)

## W

white\_background() (CImagine property), [21](#)

