

dreamtools

Release 1.0.3

dreamgeeker

déc. 20, 2020

1	Dreamtools	1
1.1	Installation	1
1.2	Configuration	1
1.3	Crédits	1
2	Gestion configuration	3
2.1	Repertoires par défaut	3
2.2	Class CFBases	3
2.3	Class CReponder	4
2.4	Class CTracker	4
3	Gestions des logs	5
4	Librairie de fonctions	7
4.1	Module de fonctions basiques	7
4.2	Module complémentaire	12
5	Module de validation	15
6	Gestion de mailing	17
6.1	Pré-Requis	17
6.2	Class CMailer	17
7	Traitement d'image	19
8	Module de cryptage	21
9	Module de Gestion des date	23
9.1	Constantes globales	23
9.2	Fonctions	23
	Index des modules Python	29
	Index	31

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 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

Paramètres **code** (*str*) – référence du formulaire

Renvoie parametres de validation

Type renvoyé dict

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` = “àâãäåèéëîïôöðõùüÿñç”
- `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
 - **l** (*list*) – liste

addhex (h, v)

Additionne une valeur hexadecimal

- Paramètres**
- **h** (*str*) – valeur hexadecimal
 - **v** (*str*) – 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 caractère 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 caractère 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*

dictlist (*k, v, d*)

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

Paramètres

- **k** (*str*) – clé dictionnaire
- **v** – valeur à ajouter
- **list[] p_dic** (*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']}
```

dir_parent (*path*)

Renvoie du repertoire parent

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

Type renvoyé str

dir_parser (*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 dir_parser(directory, pattern):
...     print(path_file)
'C:\Users\public\Documents\fichier.txt'
'C:\Users\public\Documents\autre_fichier.txt'
```

dir_worked ()

Répertoire d'exécution

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

• `v` (*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_id*, *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 distionnaire

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 ficchier <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

presend (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

class CImagine (*src, dest, with_ext=False*)

protected (*artist*)

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

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

Redimensionnement de l'image au format jpg

Paramètres • **int** (*tuple(int,)*) – si indiqué, (w,h) de redimensionnement

• **mn, default 250** (*int*) – taille maximum (carré), optional

• **mx, default 250** (*int*) – taille maximum de l'image, optional

Renvoie

thumbed (*s=None*)

Thumb Image

Paramètres int **s** (*tuple(int,)*) – taille image, default (200, 200)

property white_background

Fond blanc, utilisé pour remplacer un fond transparent

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

Param datetime dte: date de référence

Paramètres 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*)

Ddate 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** (*date*) – 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

dtediff (dte, dteb)

Calcul du nombre de jours entre deux dates

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

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

Type renvoyé int

dtets (*dte=None*)

Conversion date - timestamp

Paramètres **dt** (*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 (*s_iso*)

Conversion str_iso - date

Format ISO : YYYY-MM-DDTHH:MN :param p_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
- **p_iso** (*bool*) – "iso" = Format iso | "ts" = timestamp | None = datetime

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

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

Conversion string - date

Paramètres

- **dt** (*str*) – date
- **pt, default** **'%d-%m-%Y %H:%M:%S'** (*str*) – 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 depart + 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, 21
- dreamtools.features, 12
- dreamtools.imagine, 19
- dreamtools.logmng, 5
- dreamtools.mailbot, 17
- dreamtools.tools, 7
- dreamtools.validata, 15

A

add_list() (dans le module dreamtools.tools), 8
 addhex() (dans le module dreamtools.tools), 8
 aleatoire() (dans le module dreamtools.tools), 8
 app_cfg() (méthode statique CFGBases), 3

C

categorie_lib() (méthode statique CFGBases), 3
 CError, 5
 CFGBases (classe dans dreamtools.cfgmng), 3
 check_password() (dans le module dreamtools.-tools), 8
 check_post_data() (méthode statique Validata), 15
 CImagine (classe dans dreamtools.imagine), 19
 clean_allspace() (dans le module dreamtools.-tools), 8
 clean_coma() (dans le module dreamtools.tools), 9
 clean_dir() (dans le module dreamtools.tools), 9
 clean_master() (dans le module dreamtools.-tools), 9
 clean_space() (dans le module dreamtools.tools), 9
 CMailer (classe dans dreamtools.mailbot), 17
 code_maker() (dans le module dreamtools.tools), 9
 comphex() (dans le module dreamtools.tools), 9

D

date_add_workday() (dans le module dreamtools.dtemng), 23
 date_dayed() (dans le module dreamtools.dtemng), 23
 date_rss() (dans le module dreamtools.dtemng), 23
 dateadd() (dans le module dreamtools.dtemng), 23

datepaques() (dans le module dreamtools.dtemng), 24
 datestr() (dans le module dreamtools.dtemng), 24
 datetime_from_local_to_utc() (dans le module dreamtools.dtemng), 24
 datetime_from_utc_to_local() (dans le module dreamtools.dtemng), 24
 day_in_hour() (dans le module dreamtools.dtemng), 24
 day_in_sec() (dans le module dreamtools.dtemng), 24
 dictlist() (dans le module dreamtools.tools), 10
 dir_parent() (dans le module dreamtools.tools), 10
 dir_parser() (dans le module dreamtools.tools), 10
 dir_worked() (dans le module dreamtools.tools), 10
 dreamtools.cfgmng
 module, 3
 dreamtools.dtemng
 module, 21
 dreamtools.features
 module, 12
 dreamtools.imagine
 module, 19
 dreamtools.logmng
 module, 5
 dreamtools.mailbot
 module, 17
 dreamtools.tools
 module, 7
 dreamtools.validata
 module, 15
 dtediff() (dans le module dreamtools.dtemng), 24
 dtets() (dans le module dreamtools.dtemng), 25

F

`file_exists()` (dans le module `dreamtools.tools`), 10
`file_ext()` (dans le module `dreamtools.tools`), 10
`fullmonth()` (dans le module `dreamtools.dtemng`), 25

G

`get_date()` (dans le module `dreamtools.dtemng`), 25
`get_time()` (dans le module `dreamtools.dtemng`), 25
`get_weeks_num()` (dans le module `dreamtools.dtemng`), 25

I

`inttohex()` (dans le module `dreamtools.tools`), 11
`is_workday()` (dans le module `dreamtools.dtemng`), 25
`isotodate()` (dans le module `dreamtools.dtemng`), 25

J

`jours_feries()` (dans le module `dreamtools.dtemng`), 25

L

`logs_cfg()` (méthode statique `CFGBases`), 4

M

`mailing_lib()` (méthode statique `CFGBases`), 4
`maintenant()` (dans le module `dreamtools.dtemng`), 26
`makedirs()` (dans le module `dreamtools.tools`), 11
module
 `dreamtools.cfgmng`, 3
 `dreamtools.dtemng`, 21
 `dreamtools.features`, 12
 `dreamtools.imagine`, 19
 `dreamtools.logmng`, 5
 `dreamtools.mailbot`, 17
 `dreamtools.tools`, 7
 `dreamtools.validata`, 15

N

`normalisation()` (méthode `Validata`), 15
`normalizer()` (méthode statique `CFGBases`), 4

P

`path_build()` (dans le module `dreamtools.tools`), 11

`plain_hex()` (dans le module `dreamtools.tools`), 11
`plain_zero()` (dans le module `dreamtools.tools`), 11
`pop_dic()` (dans le module `dreamtools.tools`), 11
`presend()` (méthode `CMailer`), 17
`print_err()` (dans le module `dreamtools.tools`), 12
`protected()` (méthode `CImagine`), 19
`pwd_maker()` (dans le module `dreamtools.tools`), 12

R

`remove_file()` (dans le module `dreamtools.tools`), 12
`resize()` (méthode `CImagine`), 19

S

`set_timezone()` (dans le module `dreamtools.dtemng`), 26
`str_dic()` (dans le module `dreamtools.tools`), 12
`strdate()` (dans le module `dreamtools.dtemng`), 26
`string_me()` (dans le module `dreamtools.tools`), 12

T

`test_http_link()` (dans le module `dreamtools.features`), 12
`thumbed()` (méthode `CImagine`), 19
`timeadd()` (dans le module `dreamtools.dtemng`), 26
`today()` (dans le module `dreamtools.dtemng`), 26
`tsdate()` (dans le module `dreamtools.dtemng`), 27
`tstring()` (dans le module `dreamtools.dtemng`), 27
`TYPE_IMG_GIF` (dans le module `dreamtools.imagine`), 19

U

`url_join()` (dans le module `dreamtools.features`), 13
`utcnw_iso()` (dans le module `dreamtools.dtemng`), 27
`utcnw_ts()` (dans le module `dreamtools.dtemng`), 27

V

`Validata` (classe dans `dreamtools.validata`), 15
`validation()` (méthode `Validata`), 15
`validator()` (méthode statique `CFGBases`), 4

W

`white_background()` (`CImagine` property), 19