API Documentation

API Documentation

December 10, 2015

Contents

1 Module codecs

codecs – Python Codec Registry, API and helpers.

Written by Marc-Andre Lemburg (mal@lemburg.com).

(c) Copyright CNRI, All Rights Reserved. NO WARRANTY.

1.1 Functions

EncodedFile(*file*, *data_encoding*, *file_encoding*=`None`, *errors*=`'strict'`)

Return a wrapped version of file which provides transparent encoding translation.

Strings written to the wrapped file are interpreted according to the given data_encoding and then written to the original file as string using file_encoding. The intermediate encoding will usually be Unicode but depends on the specified codecs.

Strings are read from the file using file_encoding and then passed back to the caller as string using data_encoding.

If file_encoding is not given, it defaults to data_encoding.

errors may be given to define the error handling. It defaults to 'strict' which causes ValueErrors to be raised in case an encoding error occurs.

The returned wrapped file object provides two extra attributes .data_encoding and .file_encoding which reflect the given parameters of the same name. The attributes can be used for introspection by Python programs.

---

ignore_errors(*...*)

Implements the 'ignore' error handling, which ignores malformed data and continues.

---

lookup(*encoding*)

Looks up a codec tuple in the Python codec registry and returns a CodecInfo object.

Return Value
    CodecInfo

---

lookup_error(*errors*)

Return the error handler for the specified error handling name or raise a LookupError, if no handler exists under this name.

Return Value
    handler

---

open(*filename*, *mode*='rb', *encoding*=None, *errors*='strict', *buffering*=1)

Open an encoded file using the given mode and return a wrapped version providing transparent encoding/decoding.

Note: The wrapped version will only accept the object format defined by the codecs, i.e. Unicode objects for most builtin codecs. Output is also codec dependent and will usually be Unicode as well.

Files are always opened in binary mode, even if no binary mode was specified. This is done to avoid data loss due to encodings using 8-bit values. The default file mode is 'rb' meaning to open the file in binary read mode.

encoding specifies the encoding which is to be used for the file.

errors may be given to define the error handling. It defaults to 'strict' which causes ValueErrors to be raised in case an encoding error occurs.

buffering has the same meaning as for the builtin open() API. It defaults to line buffered.

The returned wrapped file object provides an extra attribute .encoding which allows querying the used encoding. This attribute is only available if an encoding was specified as parameter.

---

register(*search_function*)

Register a codec search function. Search functions are expected to take one argument, the encoding name in all lower case letters, and return a tuple of functions (encoder, decoder, stream_reader, stream_writer) (or a CodecInfo object).

---

register_error(*errors*, *handler*)

Register the specified error handler under the name errors. handler must be a callable object, that will be called with an exception instance containing information about the location of the encoding/decoding error and must return a (replacement, new position) tuple.

---

replace_errors(*...*)

Implements the 'replace' error handling, which replaces malformed data with a replacement marker.

---

strict_errors(*...*)

Implements the 'strict' error handling, which raises a UnicodeError on coding errors.

---

xmlcharrefreplace_errors(*...*)

Implements the 'xmlcharrefreplace' error handling, which replaces an unencodable character with the appropriate XML character reference.

1.2 Variables

Name	Description
BOM	**Value:** '\xff\xfe'
BOM32_BE	**Value:** '\xfe\xff'
BOM32_LE	**Value:** '\xff\xfe'
BOM64_BE	**Value:** '\x00\x00\xfe\xff'
BOM64_LE	**Value:** '\xff\xfe\x00\x00'
BOM_BE	**Value:** '\xfe\xff'
BOM_LE	**Value:** '\xff\xfe'
BOM_UTF16	**Value:** '\xff\xfe'
BOM_UTF16_BE	**Value:** '\xfe\xff'
BOM_UTF16_LE	**Value:** '\xff\xfe'
BOM_UTF32	**Value:** '\xff\xfe\x00\x00'
BOM_UTF32_BE	**Value:** '\x00\x00\xfe\xff'
BOM_UTF32_LE	**Value:** '\xff\xfe\x00\x00'
BOM_UTF8	**Value:** '\xef\xbb\xbf'

2 Module credit

2.1 Functions

---

credit(*text*, *font*, *color*, *fps*=40, *callback*=None)

Créditos rolando verticalmente

Parameters

    `text:` Texto

    `font:` Fonte

    `color:` Cor

---

credit_from_file(*file_name*, *font*, *color*, *fps*=40, *callback*=None)

2.2 Variables

Name	Description
__package__	**Value:** None

3 Module math

This module is always available. It provides access to the mathematical functions defined by the C standard.

3.1 Functions

acos(x)

Return the arc cosine (measured in radians) of x.

acosh(x)

Return the hyperbolic arc cosine (measured in radians) of x.

asin(x)

Return the arc sine (measured in radians) of x.

asinh(x)

Return the hyperbolic arc sine (measured in radians) of x.

atan(x)

Return the arc tangent (measured in radians) of x.

atan2(y, x)

Return the arc tangent (measured in radians) of y/x. Unlike atan(y/x), the signs of both x and y are considered.

atanh(x)

Return the hyperbolic arc tangent (measured in radians) of x.

ceil(x)

Return the ceiling of x as a float. This is the smallest integral value >= x.

copysign(x, y)

Return x with the sign of y.

cos(x)

Return the cosine of x (measured in radians).

cosh(x)

Return the hyperbolic cosine of x.

degrees(*x*)

Convert angle x from radians to degrees.

---

erf(*x*)

Error function at x.

---

erfc(*x*)

Complementary error function at x.

---

exp(*x*)

Return e raised to the power of x.

---

expm1(*x*)

Return exp(x)-1. This function avoids the loss of precision involved in the direct evaluation of exp(x)-1 for small x.

---

fabs(*x*)

Return the absolute value of the float x.

---

factorial(*x*)

Find x!. Raise a ValueError if x is negative or non-integral.

Return Value
    Integral

---

floor(*x*)

Return the floor of x as a float. This is the largest integral value <= x.

---

fmod(*x*, *y*)

Return fmod(x, y), according to platform C. x % y may differ.

---

frexp(*x*)

Return the mantissa and exponent of x, as pair (m, e). m is a float and e is an int, such that x = m * 2.**e. If x is 0, m and e are both 0. Else 0.5 <= abs(m) < 1.0.

---

fsum(*iterable*)

Return an accurate floating point sum of values in the iterable. Assumes IEEE-754 floating point arithmetic.

---

gamma(*x*)

Gamma function at x.

hypot(*x*, *y*)

Return the Euclidean distance, sqrt(x*x + y*y).

---

isinf(*x*)

Check if float x is infinite (positive or negative).

Return Value
    bool

---

isnan(*x*)

Check if float x is not a number (NaN).

Return Value
    bool

---

ldexp(*x*, *i*)

Return x * (2**i).

---

lgamma(*x*)

Natural logarithm of absolute value of Gamma function at x.

---

log(*x*, *base*=...)

Return the logarithm of x to the given base. If the base not specified, returns the natural logarithm (base e) of x.

---

log10(*x*)

Return the base 10 logarithm of x.

---

log1p(*x*)

Return the natural logarithm of 1+x (base e). The result is computed in a way which is accurate for x near zero.

---

modf(*x*)

Return the fractional and integer parts of x. Both results carry the sign of x and are floats.

---

pow(*x*, *y*)

Return x**y (x to the power of y).

---

radians(*x*)

Convert angle x from degrees to radians.

---

sin(*x*)

Return the sine of x (measured in radians).

---

sinh(x)

Return the hyperbolic sine of x.

---

sqrt(x)

Return the square root of x.

---

tan(x)

Return the tangent of x (measured in radians).

---

tanh(x)

Return the hyperbolic tangent of x.

---

trunc(...)

trunc(x:Real) -> Integral

Truncates x to the nearest Integral toward 0. Uses the __trunc__ magic method.

3.2 Variables

Name	Description
__package__	**Value:** None
e	**Value:** 2.71828182846
pi	**Value:** 3.14159265359

4 Module os

OS routines for NT or Posix depending on what system we're on.

This exports:

- all functions from posix, nt, os2, or ce, e.g. unlink, stat, etc.
- os.path is one of the modules posixpath, or ntpath
- os.name is 'posix', 'nt', 'os2', 'ce' or 'riscos'
- os.curdir is a string representing the current directory ('.' or ':')
- os.pardir is a string representing the parent directory ('..' or '::')
- os.sep is the (or a most common) pathname separator ('/' or ':' or '\\')
- os.extsep is the extension separator ('.' or '/')
- os.altsep is the alternate pathname separator (None or '/')
- os.pathsep is the component separator used in $PATH etc
- os.linesep is the line separator in text files ('\r' or '\n' or '\r\n')
- os.defpath is the default search path for executables
- os.devnull is the file path of the null device ('/dev/null', etc.)

Programs that import and use 'os' stand a better chance of being portable between different platforms. Of course, they must then only use functions that are defined by all platforms (e.g., unlink and opendir), and leave all pathname manipulation to os.path (e.g., split and join).

4.1 Functions

WCOREDUMP(*status*)

Return True if the process returning 'status' was dumped to a core file.

Return Value
> bool

WEXITSTATUS(*status*)

Return the process return code from 'status'.

Return Value
> integer

WIFCONTINUED(*status*)

Return True if the process returning 'status' was continued from a job control stop.

Return Value
> bool

WIFEXITED(*status*)

Return true if the process returning 'status' exited using the exit() system call.

Return Value
> bool

WIFSIGNALED(*status*)

Return True if the process returning 'status' was terminated by a signal.

Return Value
     bool

WIFSTOPPED(*status*)

Return True if the process returning 'status' was stopped.

Return Value
     bool

WSTOPSIG(*status*)

Return the signal that stopped the process that provided the 'status' value.

Return Value
     integer

WTERMSIG(*status*)

Return the signal that terminated the process that provided the 'status' value.

Return Value
     integer

---

abort()

Abort the interpreter immediately. This 'dumps core' or otherwise fails in the hardest way possible on the hosting operating system.

Return Value
     does not return!

---

access(*path*, *mode*)

Use the real uid/gid to test for access to a path. Note that most operations will use the effective uid/gid, therefore this routine can be used in a suid/sgid environment to test if the invoking user has the specified access to the path. The mode argument can be F_OK to test existence, or the inclusive-OR of R_OK, W_OK, and X_OK.

Return Value
     True if granted, False otherwise

---

chdir(*path*)

Change the current working directory to the specified path.

---

chflags(*path*, *flags*)

Set file flags.

---

chmod(*path*, *mode*)

Change the access permissions of a file.

---

chown(*path, uid, gid*)

Change the owner and group id of path to the numeric uid and gid.

---

chroot(*path*)

Change root directory to path.

---

close(*fd*)

Close a file descriptor (for low level IO).

---

closerange(*fd_low, fd_high*)

Closes all file descriptors in [fd_low, fd_high), ignoring errors.

---

confstr(*name*)

Return a string-valued system configuration variable.

Return Value
    string

---

ctermid()

Return the name of the controlling terminal for this process.

Return Value
    string

---

dup(*fd*)

Return a duplicate of a file descriptor.

Return Value
    fd2

---

dup2(*old_fd, new_fd*)

Duplicate file descriptor.

---

execl(*file, *args*)

Execute the executable file with argument list args, replacing the current process.

---

execle(*file, env, *args*)

Execute the executable file with argument list args and environment env, replacing the current process.

---

execlp(*file, *args*)

Execute the executable file (which is searched for along $PATH) with argument list args, replacing the current process.

execlpe(*file, env, *args*)

Execute the executable file (which is searched for along $PATH) with argument list args and environment env, replacing the current process.

---

execv(*path, args*)

```
Execute an executable path with arguments, replacing current process.

    path: path of executable file
    args: tuple or list of strings
```

---

execve(*path, args, env*)

```
Execute a path with arguments and environment, replacing current process.

    path: path of executable file
    args: tuple or list of arguments
    env: dictionary of strings mapping to strings
```

---

execvp(*file, args*)

Execute the executable file (which is searched for along $PATH) with argument list args, replacing the current process. args may be a list or tuple of strings.

---

execvpe(*file, args, env*)

Execute the executable file (which is searched for along $PATH) with argument list args and environment env , replacing the current process. args may be a list or tuple of strings.

---

fchdir(*fildes*)

Change to the directory of the given file descriptor. fildes must be opened on a directory, not a file.

---

fchmod(*fd, mode*)

Change the access permissions of the file given by file descriptor fd.

---

fchown(*fd, uid, gid*)

Change the owner and group id of the file given by file descriptor fd to the numeric uid and gid.

---

fdopen(*fd, mode=*'**r**'*, bufsize=...*)

Return an open file object connected to a file descriptor.

Return Value
     file_object

fork()

Fork a child process. Return 0 to child process and PID of child to parent process.

Return Value
    pid

---

forkpty()

Fork a new process with a new pseudo-terminal as controlling tty.

Like fork(), return 0 as pid to child process, and PID of child to parent. To both, return fd of newly opened pseudo-terminal.

Return Value
    (pid, master_fd)

---

fpathconf(*fd, name*)

Return the configuration limit name for the file descriptor fd. If there is no limit, return -1.

Return Value
    integer

---

fstat(*fd*)

Like stat(), but for an open file descriptor.

Return Value
    stat result

---

fstatvfs(*fd*)

Perform an fstatvfs system call on the given fd.

Return Value
    statvfs result

---

fsync(*fildes*)

force write of file with filedescriptor to disk.

---

ftruncate(*fd, length*)

Truncate a file to a specified length.

---

getcwd()

Return a string representing the current working directory.

Return Value
    path

getcwdu()

Return a unicode string representing the current working directory.

Return Value
    path

---

getegid()

Return the current process's effective group id.

Return Value
    egid

---

getenv(*key*, *default=*`None`)

Get an environment variable, return None if it doesn't exist. The optional second argument can specify an alternate default.

---

geteuid()

Return the current process's effective user id.

Return Value
    euid

---

getgid()

Return the current process's group id.

Return Value
    gid

---

getgroups()

Return list of supplemental group IDs for the process.

Return Value
    list of group IDs

---

getloadavg()

Return the number of processes in the system run queue averaged over the last 1, 5, and 15 minutes or raises OSError if the load average was unobtainable

Return Value
    (float, float, float)

---

getlogin()

Return the actual login name.

Return Value
    string

getpgid(*pid*)

Call the system call getpgid().

Return Value
    pgid

---

getpgrp()

Return the current process group id.

Return Value
    pgrp

---

getpid()

Return the current process id

Return Value
    pid

---

getppid()

Return the parent's process id.

Return Value
    ppid

---

getsid(*pid*)

Call the system call getsid().

Return Value
    sid

---

getuid()

Return the current process's user id.

Return Value
    uid

---

initgroups(*username*, *gid*)

Call the system initgroups() to initialize the group access list with all of the groups of which the specified username is a member, plus the specified group id.

Return Value
    None

---

isatty(*fd*)

Return True if the file descriptor 'fd' is an open file descriptor connected to the slave end of a terminal.

Return Value
    bool

kill(*pid*, *sig*)

Kill a process with a signal.

---

killpg(*pgid*, *sig*)

Kill a process group with a signal.

---

lchflags(*path*, *flags*)

Set file flags. This function will not follow symbolic links.

---

lchmod(*path*, *mode*)

Change the access permissions of a file. If path is a symlink, this affects the link itself rather than the target.

---

lchown(*path*, *uid*, *gid*)

Change the owner and group id of path to the numeric uid and gid. This function will not follow symbolic links.

---

link(*src*, *dst*)

Create a hard link to a file.

---

listdir(*path*)

```
Return a list containing the names of the entries in the directory.

    path: path of directory to list

The list is in arbitrary order.  It does not include the special
entries '.' and '..' even if they are present in the directory.
```

Return Value
  list_of_strings

---

lseek(*fd*, *pos*, *how*)

Set the current position of a file descriptor. Return the new cursor position in bytes, starting from the beginning.

Return Value
  newpos

---

lstat(*path*)

Like stat(path), but do not follow symbolic links.

Return Value
  stat result

major(*device*)

Extracts a device major number from a raw device number.

Return Value
    major number

---

makedev(*major*, *minor*)

Composes a raw device number from the major and minor device numbers.

Return Value
    device number

---

makedirs(*path*, *mode*=0777)

Super-mkdir; create a leaf directory and all intermediate ones. Works like mkdir, except that any intermediate path segment (not just the rightmost) will be created if it does not exist. This is recursive.

---

minor(*device*)

Extracts a device minor number from a raw device number.

Return Value
    minor number

---

mkdir(*path*, *mode*=0777)

Create a directory.

---

mkfifo(*filename*, *mode*=0666)

Create a FIFO (a POSIX named pipe).

---

mknod(*filename*, *mode*=0600, *device*=...)

Create a filesystem node (file, device special file or named pipe) named filename. mode specifies both the permissions to use and the type of node to be created, being combined (bitwise OR) with one of S_IFREG, S_IFCHR, S_IFBLK, and S_IFIFO. For S_IFCHR and S_IFBLK, device defines the newly created device special file (probably using os.makedev()), otherwise it is ignored.

---

nice(*inc*)

Decrease the priority of process by inc and return the new priority.

Return Value
    new_priority

---

open(*filename*, *flag*, *mode*=0777)

Open a file (for low level IO).

Return Value
    fd

---

openpty()

Open a pseudo-terminal, returning open fd's for both master and slave end.

Return Value
  (master_fd, slave_fd)

---

pathconf(*path*, *name*)

Return the configuration limit name for the file or directory path. If there is no limit, return -1.

Return Value
  integer

---

pipe()

Create a pipe.

Return Value
  (read_end, write_end)

---

popen(*command*, *mode*='r' , *bufsize*=...)

Open a pipe to/from a command returning a file object.

Return Value
  pipe

---

popen2(*cmd*, *mode*='t', *bufsize*=-1)

Execute the shell command 'cmd' in a sub-process. On UNIX, 'cmd' may be a sequence, in which case arguments will be passed directly to the program without shell intervention (as with os.spawnv()). If 'cmd' is a string it will be passed to the shell (as with os.system()). If 'bufsize' is specified, it sets the buffer size for the I/O pipes. The file objects (child_stdin, child_stdout) are returned.

---

popen3(*cmd*, *mode*='t', *bufsize*=-1)

Execute the shell command 'cmd' in a sub-process. On UNIX, 'cmd' may be a sequence, in which case arguments will be passed directly to the program without shell intervention (as with os.spawnv()). If 'cmd' is a string it will be passed to the shell (as with os.system()). If 'bufsize' is specified, it sets the buffer size for the I/O pipes. The file objects (child_stdin, child_stdout, child_stderr) are returned.

---

popen4(*cmd*, *mode*='t', *bufsize*=-1)

Execute the shell command 'cmd' in a sub-process. On UNIX, 'cmd' may be a sequence, in which case arguments will be passed directly to the program without shell intervention (as with os.spawnv()). If 'cmd' is a string it will be passed to the shell (as with os.system()). If 'bufsize' is specified, it sets the buffer size for the I/O pipes. The file objects (child_stdin, child_stdout_stderr) are returned.

putenv(*key, value*)

Change or add an environment variable.

---

read(*fd, buffersize*)

Read a file descriptor.

Return Value
    string

---

readlink(*path*)

Return a string representing the path to which the symbolic link points.

Return Value
    path

---

remove(*path*)

Remove a file (same as unlink(path)).

---

removedirs(*path*)

Super-rmdir; remove a leaf directory and all empty intermediate ones. Works like rmdir except that, if the leaf directory is successfully removed, directories corresponding to rightmost path segments will be pruned away until either the whole path is consumed or an error occurs. Errors during this latter phase are ignored – they generally mean that a directory was not empty.

---

rename(*old, new*)

Rename a file or directory.

---

renames(*old, new*)

Super-rename; create directories as necessary and delete any left empty. Works like rename, except creation of any intermediate directories needed to make the new pathname good is attempted first. After the rename, directories corresponding to rightmost path segments of the old name will be pruned way until either the whole path is consumed or a nonempty directory is found.

Note: this function can fail with the new directory structure made if you lack permissions needed to unlink the leaf directory or file.

---

rmdir(*path*)

Remove a directory.

---

setegid(*gid*)

Set the current process's effective group id.

---

seteuid(*uid*)

Set the current process's effective user id.

---

setgid(*gid*)

Set the current process's group id.

---

setgroups(*list*)

Set the groups of the current process to list.

---

setpgid(*pid, pgrp*)

Call the system call setpgid().

---

setpgrp()

Make this process the process group leader.

---

setregid(*rgid, egid*)

Set the current process's real and effective group ids.

---

setreuid(*ruid, euid*)

Set the current process's real and effective user ids.

---

setsid()

Call the system call setsid().

---

setuid(*uid*)

Set the current process's user id.

---

spawnl(*mode, file, *args*)

Execute file with arguments from args in a subprocess. If mode == P_NOWAIT return the pid of the process. If mode == P_WAIT return the process's exit code if it exits normally; otherwise return -SIG, where SIG is the signal that killed it.

Return Value
    integer

---

spawnle(*mode, file, env, *args*)

Execute file with arguments from args in a subprocess with the supplied environment. If mode == P_NOWAIT return the pid of the process. If mode == P_WAIT return the process's exit code if it exits normally; otherwise return -SIG, where SIG is the signal that killed it.

Return Value
    integer

---

spawnlp(*mode, file, *args*)

Execute file (which is looked for along $PATH) with arguments from args in a subprocess with the supplied environment. If mode == P_NOWAIT return the pid of the process. If mode == P_WAIT return the process's exit code if it exits normally; otherwise return -SIG, where SIG is the signal that killed it.

Return Value
    integer

---

spawnlpe(*mode, file, env, *args*)

Execute file (which is looked for along $PATH) with arguments from args in a subprocess with the supplied environment. If mode == P_NOWAIT return the pid of the process. If mode == P_WAIT return the process's exit code if it exits normally; otherwise return -SIG, where SIG is the signal that killed it.

Return Value
    integer

---

spawnv(*mode, file, args*)

Execute file with arguments from args in a subprocess. If mode == P_NOWAIT return the pid of the process. If mode == P_WAIT return the process's exit code if it exits normally; otherwise return -SIG, where SIG is the signal that killed it.

Return Value
    integer

---

spawnve(*mode, file, args, env*)

Execute file with arguments from args in a subprocess with the specified environment. If mode == P_NOWAIT return the pid of the process. If mode == P_WAIT return the process's exit code if it exits normally; otherwise return -SIG, where SIG is the signal that killed it.

Return Value
    integer

---

spawnvp(*mode, file, args*)

Execute file (which is looked for along $PATH) with arguments from args in a subprocess. If mode == P_NOWAIT return the pid of the process. If mode == P_WAIT return the process's exit code if it exits normally; otherwise return -SIG, where SIG is the signal that killed it.

Return Value
    integer

spawnvpe(*mode, file, args, env*)

Execute file (which is looked for along $PATH) with arguments from args in a subprocess with the supplied environment. If mode == P_NOWAIT return the pid of the process. If mode == P_WAIT return the process's exit code if it exits normally; otherwise return -SIG, where SIG is the signal that killed it.

Return Value
    integer

---

stat(*path*)

Perform a stat system call on the given path.

Return Value
    stat result

---

stat_float_times(*newval=...*)

Determine whether os.[lf]stat represents time stamps as float objects. If newval is True, future calls to stat() return floats, if it is False, future calls return ints. If newval is omitted, return the current setting.

Return Value
    oldval

---

statvfs(*path*)

Perform a statvfs system call on the given path.

Return Value
    statvfs result

---

strerror(*code*)

Translate an error code to a message string.

Return Value
    string

---

symlink(*src, dst*)

Create a symbolic link pointing to src named dst.

---

sysconf(*name*)

Return an integer-valued system configuration variable.

Return Value
    integer

---

system(*command*)

Execute the command (a string) in a subshell.

Return Value
    exit_status

tcgetpgrp(*fd*)

Return the process group associated with the terminal given by a fd.

Return Value
> pgid

tcsetpgrp(*fd*, *pgid*)

Set the process group associated with the terminal given by a fd.

tempnam(*dir=*..., *prefix=*...)

Return a unique name for a temporary file. The directory and a prefix may be specified as strings; they may be omitted or None if not needed.

Return Value
> string

times()

Return a tuple of floating point numbers indicating process times.

Return Value
> (utime, stime, cutime, cstime, elapsed_time)

tmpfile()

Create a temporary file with no directory entries.

Return Value
> file object

tmpnam()

Return a unique name for a temporary file.

Return Value
> string

ttyname(*fd*)

Return the name of the terminal device connected to 'fd'.

Return Value
> string

umask(*new_mask*)

Set the current numeric umask and return the previous umask.

Return Value
> old_mask

---

uname()

Return a tuple identifying the current operating system.

Return Value
> (sysname, nodename, release, version, machine)

---

unlink(*path*)

Remove a file (same as remove(path)).

---

unsetenv(*key*)

Delete an environment variable.

---

urandom(*n*)

Return n random bytes suitable for cryptographic use.

Return Value
> str

---

utime(*...*)

utime(path, (atime, mtime)) utime(path, None)

Set the access and modified time of the file to the given values. If the second form is used, set the access and modified times to the current time.

---

wait()

Wait for completion of a child process.

Return Value
> (pid, status)

---

wait3(*options*)

Wait for completion of a child process.

Return Value
> (pid, status, rusage)

---

wait4(*pid*, *options*)

Wait for completion of a given child process.

Return Value
> (pid, status, rusage)

---

waitpid(*pid*, *options*)

Wait for completion of a given child process.

Return Value
> (pid, status)

walk(*top*, *topdown*=True, *onerror*=None, *followlinks*=False)

```
Directory tree generator.

For each directory in the directory tree rooted at top (including top
itself, but excluding '.' and '..'), yields a 3-tuple

    dirpath, dirnames, filenames

dirpath is a string, the path to the directory.  dirnames is a list of
the names of the subdirectories in dirpath (excluding '.' and '..').
filenames is a list of the names of the non-directory files in dirpath.
Note that the names in the lists are just names, with no path components.
To get a full path (which begins with top) to a file or directory in
dirpath, do os.path.join(dirpath, name).

If optional arg 'topdown' is true or not specified, the triple for a
directory is generated before the triples for any of its subdirectories
(directories are generated top down).  If topdown is false, the triple
for a directory is generated after the triples for all of its
subdirectories (directories are generated bottom up).

When topdown is true, the caller can modify the dirnames list in-place
(e.g., via del or slice assignment), and walk will only recurse into the
subdirectories whose names remain in dirnames; this can be used to prune the
search, or to impose a specific order of visiting.  Modifying dirnames when
topdown is false is ineffective, since the directories in dirnames have
already been generated by the time dirnames itself is generated. No matter
the value of topdown, the list of subdirectories is retrieved before the
tuples for the directory and its subdirectories are generated.

By default errors from the os.listdir() call are ignored.  If
optional arg 'onerror' is specified, it should be a function; it
will be called with one argument, an os.error instance.  It can
report the error to continue with the walk, or raise the exception
to abort the walk.  Note that the filename is available as the
filename attribute of the exception object.

By default, os.walk does not follow symbolic links to subdirectories on
systems that support them.  In order to get this functionality, set the
optional argument 'followlinks' to true.

Caution:  if you pass a relative pathname for top, don't change the
current working directory between resumptions of walk.  walk never
changes the current directory, and assumes that the client doesn't
either.

Example:

import os
from os.path import join, getsize
for root, dirs, files in os.walk('python/Lib/email'):
    print root, "consumes",
    print sum([getsize(join(root, name)) for name in files]),
    print "bytes in", len(files), "non-directory files"
    if 'CVS' in dirs:
        dirs.remove('CVS')  # don't visit CVS directories
```

---

write(*fd, string*)

Write a string to a file descriptor.

Return Value
    byteswritten

4.2   Variables

Name	Description
EX_CANTCREAT	**Value:** 73
EX_CONFIG	**Value:** 78
EX_DATAERR	**Value:** 65
EX_IOERR	**Value:** 74
EX_NOHOST	**Value:** 68
EX_NOINPUT	**Value:** 66
EX_NOPERM	**Value:** 77
EX_NOUSER	**Value:** 67
EX_OK	**Value:** 0
EX_OSERR	**Value:** 71
EX_OSFILE	**Value:** 72
EX_PROTOCOL	**Value:** 76
EX_SOFTWARE	**Value:** 70
EX_TEMPFAIL	**Value:** 75
EX_UNAVAILABLE	**Value:** 69
EX_USAGE	**Value:** 64
F_OK	**Value:** 0
NGROUPS_MAX	**Value:** 16
O_APPEND	**Value:** 8
O_ASYNC	**Value:** 64
O_CREAT	**Value:** 512
O_DIRECTORY	**Value:** 1048576
O_DSYNC	**Value:** 4194304
O_EXCL	**Value:** 2048
O_EXLOCK	**Value:** 32
O_NDELAY	**Value:** 4
O_NOCTTY	**Value:** 131072
O_NOFOLLOW	**Value:** 256
O_NONBLOCK	**Value:** 4
O_RDONLY	**Value:** 0
O_RDWR	**Value:** 2
O_SHLOCK	**Value:** 16
O_SYNC	**Value:** 128
O_TRUNC	**Value:** 1024
O_WRONLY	**Value:** 1
R_OK	**Value:** 4
SEEK_CUR	**Value:** 1
SEEK_END	**Value:** 2
SEEK_SET	**Value:** 0
TMP_MAX	**Value:** 308915776
WCONTINUED	**Value:** 16

Name	Description
WNOHANG	**Value:** `1`
WUNTRACED	**Value:** `2`
W_OK	**Value:** `2`
X_OK	**Value:** `1`
altsep	**Value:** `None`
confstr_names	**Value:** `{'CS_PATH': 1,` `'CS_XBS5_ILP32_OFF32_CFLAGS': 20, 'CS_XBS5...`
curdir	**Value:** `'.'`
defpath	**Value:** `':/bin:/usr/bin'`
devnull	**Value:** `'/dev/null'`
environ	**Value:** `{'GOPATH':` `'/usr/local/opt/go/libexec/bin', 'HOME':` `'/Use...`
extsep	**Value:** `'.'`
linesep	**Value:** `'\n'`
name	**Value:** `'posix'`
pardir	**Value:** `'..'`
pathconf_names	**Value:** `{'PC_ASYNC_IO': 17,` `'PC_CHOWN_RESTRICTED': 7, 'PC_FILESIZ...`
pathsep	**Value:** `':'`
sep	**Value:** `'/'`
sysconf_names	**Value:** `{'SC_2_CHAR_TERM': 20, 'SC_2_C_BIND': 18,` `'SC_2_C_DEV': 1...`

4.3 Class OSError

object ⌐

exceptions.BaseException ⌐

    exceptions.Exception ⌐

        exceptions.StandardError ⌐

            exceptions.EnvironmentError ⌐

                exceptions.OSError
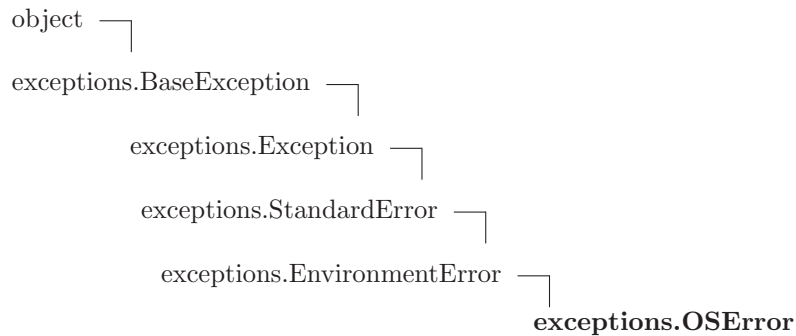
OS system call failed.

4.3.1 Methods

__init__(...)
x.__init__(...) initializes x; see help(type(x)) for signature
Overrides: object.__init__

__new__(*T*, *S*, ...)

Return Value
> a new object with type S, a subtype of T

Overrides: object.__new__

Inherited from exceptions.EnvironmentError

__reduce__(), __str__()

Inherited from exceptions.BaseException

__delattr__(), __getattribute__(), __getitem__(), __getslice__(), __repr__(), __setattr__(), __setstate__(), __unicode__()

Inherited from object

__format__(), __hash__(), __reduce_ex__(), __sizeof__(), __subclasshook__()

4.3.2 Properties

Name	Description
Inherited from exceptions.EnvironmentError	
errno, filename, strerror	
Inherited from exceptions.BaseException	
args, message	
Inherited from object	
__class__	

4.4 Class stat_result

object ─┐
        posix.stat_result

```
stat_result: Result from stat or lstat.

This object may be accessed either as a tuple of
  (mode, ino, dev, nlink, uid, gid, size, atime, mtime, ctime)
or via the attributes st_mode, st_ino, st_dev, st_nlink, st_uid, and so on.

Posix/windows: If your platform supports st_blksize, st_blocks, st_rdev,
or st_flags, they are available as attributes only.

See os.stat for more information.
```

4.4.1 Methods

__add__(*x*, *y*)
x+y

__contains__(*x*, *y*)
y in x

__eq__(*x*, *y*)
x==y

__ge__(*x*, *y*)
x>=y

__getitem__(*x*, *y*)
x[y]

__getslice__(*x*, *i*, *j*)
x[i:j]
Use of negative indices is not supported.

__gt__(*x*, *y*)
x>y

__hash__(*x*)
hash(x)
Overrides: object.__hash__

__le__(*x*, *y*)
x<=y

__len__(*x*)
len(x)

__lt__(*x, y*)

x<y

__mul__(*x, n*)

x*n

__ne__(*x, y*)

x!=y

__new__(*T, S, ...*)

Return Value
    a new object with type S, a subtype of T

Overrides: object.__new__

__reduce__(*...*)

helper for pickle

Overrides: object.__reduce__ extit(inherited documentation)

__repr__(*x*)

repr(x)

Overrides: object.__repr__

__rmul__(*x, n*)

n*x

Inherited from object

__delattr__(), __format__(), __getattribute__(), __init__(), __reduce_ex__(), __setattr__(),
__sizeof__(), __str__(), __subclasshook__()

4.4.2   Properties

Name	Description
st_atime	time of last access
st_birthtime	time of creation
st_blksize	blocksize for filesystem I/O
st_blocks	number of blocks allocated
st_ctime	time of last change

Name	Description
st_dev	device
st_flags	user defined flags for file
st_gen	generation number
st_gid	group ID of owner
st_ino	inode
st_mode	protection bits
st_mtime	time of last modification
st_nlink	number of hard links
st_rdev	device type (if inode device)
st_size	total size, in bytes
st_uid	user ID of owner
Inherited from object	
__class__	

4.4.3 Class Variables

Name	Description
n_fields	**Value:** 19
n_sequence_fields	**Value:** 10
n_unnamed_fields	**Value:** 3

4.5 Class statvfs_result

object ⏋

       posix.statvfs_result

statvfs_result: Result from statvfs or fstatvfs.

This object may be accessed either as a tuple of
  (bsize, frsize, blocks, bfree, bavail, files, ffree, favail, flag, namemax),
or via the attributes f_bsize, f_frsize, f_blocks, f_bfree, and so on.

See os.statvfs for more information.

4.5.1 Methods

__add__(x, y)
x+y

__contains__(*x*, *y*)
y in x

__eq__(*x*, *y*)
x==y

__ge__(*x*, *y*)
x>=y

__getitem__(*x*, *y*)
x[y]

__getslice__(*x*, *i*, *j*)
x[i:j]
Use of negative indices is not supported.

__gt__(*x*, *y*)
x>y

__hash__(*x*)
hash(x)
Overrides: object.__hash__

__le__(*x*, *y*)
x<=y

__len__(*x*)
len(x)

__lt__(*x*, *y*)
x<y

__mul__(*x*, *n*)
x*n

__**ne**__(*x, y*)

---

x!=y

__**new**__(*T, S, ...*)

Return Value
    a new object with type S, a subtype of T

Overrides: object.__new__

__**reduce**__(*...*)

helper for pickle

Overrides: object.__reduce__ extit(inherited documentation)

__**repr**__(*x*)

---

repr(x)

Overrides: object.__repr__

__**rmul**__(*x, n*)

---

n*x

Inherited from object

__delattr__(), __format__(), __getattribute__(), __init__(), __reduce_ex__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

4.5.2   Properties

Name	Description
f_bavail	
f_bfree	
f_blocks	
f_bsize	
f_favail	
f_ffree	
f_files	
f_flag	
f_frsize	
f_namemax	
Inherited from object	
__class__	

4.5.3 Class Variables

Name	Description
n_fields	**Value:** 10
n_sequence_fields	**Value:** 10
n_unnamed_fields	**Value:** 0

5   Module pickle

```
Create portable serialized representations of Python objects.

See module cPickle for a (much) faster implementation.
See module copy_reg for a mechanism for registering custom picklers.
See module pickletools source for extensive comments.

Classes:

    Pickler
    Unpickler

Functions:

    dump(object, file)
    dumps(object) -> string
    load(file) -> object
    loads(string) -> object

Misc variables:

    __version__
    format_version
    compatible_formats
```

Version: $Revision: 72223 $

5.1   Functions

---

dump(*obj, file, protocol*=None)

---

dumps(*obj, protocol*=None)

---

load(*file*)

---

loads(*str*)

5.2   Variables

Name	Description
APPEND	**Value:** 'a'
APPENDS	**Value:** 'e'
BINFLOAT	**Value:** 'G'
BINGET	**Value:** 'h'
BININT	**Value:** 'J'
BININT1	**Value:** 'K'
BININT2	**Value:** 'M'
BINPERSID	**Value:** 'Q'
BINPUT	**Value:** 'q'
BINSTRING	**Value:** 'T'
BINUNICODE	**Value:** 'X'
BUILD	**Value:** 'b'
DICT	**Value:** 'd'
DUP	**Value:** '2'
EMPTY_DICT	**Value:** '}'
EMPTY_LIST	**Value:** ']'
EMPTY_TUPLE	**Value:** ')'
EXT1	**Value:** '\x82'
EXT2	**Value:** '\x83'
EXT4	**Value:** '\x84'
FALSE	**Value:** 'I00\n'
FLOAT	**Value:** 'F'
GET	**Value:** 'g'
GLOBAL	**Value:** 'c'
HIGHEST_PROTOCOL	**Value:** 2
INST	**Value:** 'i'
INT	**Value:** 'I'
LIST	**Value:** 'l'
LONG	**Value:** 'L'
LONG1	**Value:** '\x8a'
LONG4	**Value:** '\x8b'
LONG_BINGET	**Value:** 'j'
LONG_BINPUT	**Value:** 'r'
MARK	**Value:** '('
NEWFALSE	**Value:** '\x89'
NEWOBJ	**Value:** '\x81'
NEWTRUE	**Value:** '\x88'
NONE	**Value:** 'N'
OBJ	**Value:** 'o'
PERSID	**Value:** 'P'
POP	**Value:** '0'
POP_MARK	**Value:** '1'

Name	Description
PROTO	**Value:** '\x80'
PUT	**Value:** 'p'
REDUCE	**Value:** 'R'
SETITEM	**Value:** 's'
SETITEMS	**Value:** 'u'
SHORT_BINSTRING	**Value:** 'U'
STOP	**Value:** '.'
STRING	**Value:** 'S'
TRUE	**Value:** 'I01\n'
TUPLE	**Value:** 't'
TUPLE1	**Value:** '\x85'
TUPLE2	**Value:** '\x86'
TUPLE3	**Value:** '\x87'
UNICODE	**Value:** 'V'

5.3 Class PickleError

object ⌐

exceptions.BaseException ⌐

        exceptions.Exception ⌐

               pickle.PickleError
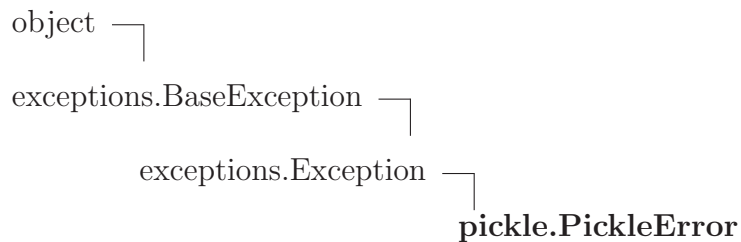
Known Subclasses: pickle.PicklingError, pickle.UnpicklingError

A common base class for the other pickling exceptions.

5.3.1 Methods

Inherited from exceptions.Exception

     __init__(), __new__()

Inherited from exceptions.BaseException

     __delattr__(), __getattribute__(), __getitem__(), __getslice__(), __reduce__(), __repr__(), __setattr__(), __setstate__(), __str__(), __unicode__()

Inherited from object

     __format__(), __hash__(), __reduce_ex__(), __sizeof__(), __subclasshook__()

5.3.2 Properties

Name	Description
Inherited from exceptions.BaseException	
args, message	
Inherited from object	
__class__	

5.4 Class Pickler

5.4.1 Methods

__init__(*self*, *file*, *protocol*=None)

This takes a file-like object for writing a pickle data stream.

The optional protocol argument tells the pickler to use the given protocol; supported protocols are 0, 1, 2. The default protocol is 0, to be backwards compatible. (Protocol 0 is the only protocol that can be written to a file opened in text mode and read back successfully. When using a protocol higher than 0, make sure the file is opened in binary mode, both when pickling and unpickling.)

Protocol 1 is more efficient than protocol 0; protocol 2 is more efficient than protocol 1.

Specifying a negative protocol version selects the highest protocol version supported. The higher the protocol used, the more recent the version of Python needed to read the pickle produced.

The file parameter must have a write() method that accepts a single string argument. It can thus be an open file object, a StringIO object, or any other custom object that meets this interface.

---

clear_memo(*self*)

Clears the pickler's "memo".

The memo is the data structure that remembers which objects the pickler has already seen, so that shared or recursive objects are pickled by reference and not by value. This method is useful when re-using picklers.

---

dump(*self*, *obj*)

Write a pickled representation of obj to the open file.

---

get(*self*, *i*, *pack=*`<built-in function pack>`)

---

memoize(*self*, *obj*)

Store an object in the memo.

---

persistent_id(*self*, *obj*)

---

put(*self*, *i*, *pack=*`<built-in function pack>`)

---

save(*self*, *obj*)

---

save_bool(*self*, *obj*)

---

save_dict(*self*, *obj*)

---

save_empty_tuple(*self*, *obj*)

---

save_float(*self*, *obj*, *pack=*`<built-in function pack>`)

---

save_global(*self*, *obj*, *name=*None, *pack=*`<built-in function pack>`)

---

save_inst(*self*, *obj*)

---

save_int(*self*, *obj*, *pack=*`<built-in function pack>`)

---

save_list(*self*, *obj*)

---

save_long(*self*, *obj*, *pack=*`<built-in function pack>`)

---

save_none(*self*, *obj*)

---

save_pers(*self*, *pid*)

---

save_reduce(*self*, *func*, *args*, *state=*None, *listitems=*None, *dictitems=*None, *obj=*None)

---

save_string(*self*, *obj*, *pack=*`<built-in function pack>`)

save_tuple(*self, obj*)

save_unicode(*self, obj, pack=*`<built-in function pack>`)

5.4.2 Class Variables

Name	Description
dispatch	**Value:** {`<type 'bool'>`: `<function` `save_bool at 0x1021342a8>`, `<typ...`

5.5 Class PicklingError

object ¬

exceptions.BaseException ¬

exceptions.Exception ¬

pickle.PickleError ¬

pickle.PicklingError
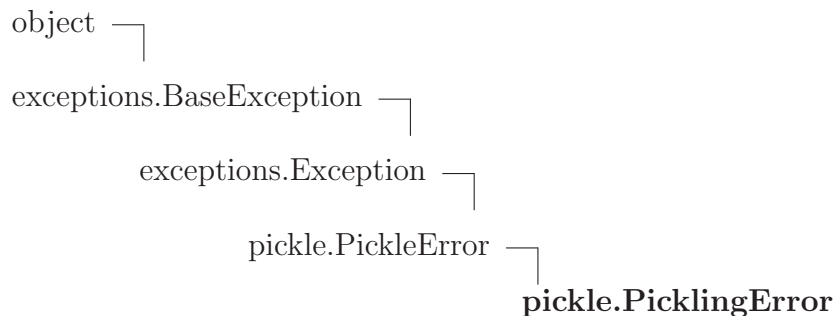
This exception is raised when an unpicklable object is passed to the dump() method.

5.5.1 Methods

Inherited from exceptions.Exception

‗‗init‗‗(), ‗‗new‗‗()

Inherited from exceptions.BaseException

‗‗delattr‗‗(), ‗‗getattribute‗‗(), ‗‗getitem‗‗(), ‗‗getslice‗‗(), ‗‗reduce‗‗(), ‗‗repr‗‗(), ‗‗setattr‗‗(), ‗‗setstate‗‗(), ‗‗str‗‗(), ‗‗unicode‗‗()

Inherited from object

‗‗format‗‗(), ‗‗hash‗‗(), ‗‗reduce_ex‗‗(), ‗‗sizeof‗‗(), ‗‗subclasshook‗‗()

5.5.2 Properties

Name	Description
Inherited from exceptions.BaseException	
args, message	
Inherited from object	
__class__	

5.6 Class Unpickler

5.6.1 Methods

__init__(*self, file*)

This takes a file-like object for reading a pickle data stream.

The protocol version of the pickle is detected automatically, so no proto argument is needed.

The file-like object must have two methods, a read() method that takes an integer argument, and a readline() method that requires no arguments. Both methods should return a string. Thus file-like object can be a file object opened for reading, a StringIO object, or any other custom object that meets this interface.

---

find_class(*self, module, name*)

---

get_extension(*self, code*)

---

load(*self*)

Read a pickled object representation from the open file.

Return the reconstituted object hierarchy specified in the file.

---

load_append(*self*)

---

load_appends(*self*)

---

load_binfloat(*self, unpack=<*`built-in function unpack`*>*)

---

load_binget(*self*)

---

load_binint(*self*)

---

load_binint1(*self*)

load_binint2(*self*)

load_binpersid(*self*)

load_binput(*self*)

load_binstring(*self*)

load_binunicode(*self*)

load_build(*self*)

load_dict(*self*)

load_dup(*self*)

load_empty_dictionary(*self*)

load_empty_list(*self*)

load_empty_tuple(*self*)

load_eof(*self*)

load_ext1(*self*)

load_ext2(*self*)

load_ext4(*self*)

load_false(*self*)

load_float(*self*)

load_get(*self*)

load_global(*self*)

load_inst(*self*)

load_int(*self*)

load_list(*self*)

load_long(*self*)

load_long1(*self*)

load_long4(*self*)

load_long_binget(*self*)

load_long_binput(*self*)

load_mark(*self*)

load_newobj(*self*)

load_none(*self*)

load_obj(*self*)

load_persid(*self*)

load_pop(*self*)

load_pop_mark(*self*)

load_proto(*self*)

load_put(*self*)

load_reduce(*self*)

load_setitem(*self*)

load_setitems(*self*)

load_short_binstring(*self*)

load_stop(*self*)

load_string(*self*)

load_true(*self*)

load_tuple(*self*)

load_tuple1(*self*)

load_tuple2(*self*)

load_tuple3(*self*)

load_unicode(*self*)

marker(*self*)

5.6.2 Class Variables

Name	Description
dispatch	**Value:** {'': <function load_eof at 0x102134b90>, '(': <function l...

5.7 Class UnpicklingError

object ─┐

exceptions.BaseException ─┐

        exceptions.Exception ─┐

                pickle.PickleError ─┐

                        pickle.UnpicklingError
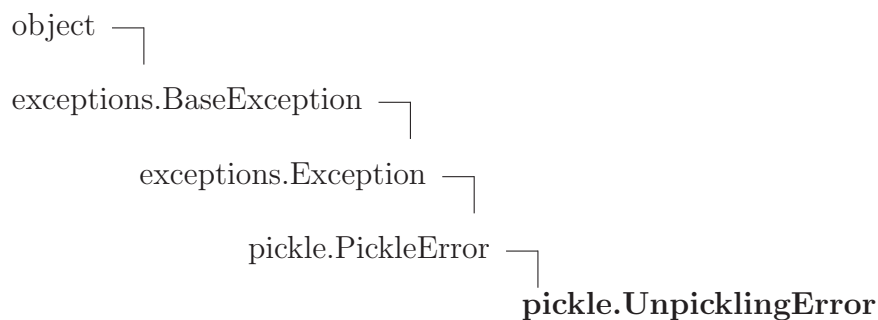
This exception is raised when there is a problem unpickling an object, such as a security violation.

Note that other exceptions may also be raised during unpickling, including (but not necessarily limited to) AttributeError, EOFError, ImportError, and IndexError.

5.7.1 Methods

Inherited from exceptions.Exception

    __init__(), __new__()

Inherited from exceptions.BaseException

    __delattr__(), __getattribute__(), __getitem__(), __getslice__(), __reduce__(), __repr__(), __setattr__(), __setstate__(), __str__(), __unicode__()

Inherited from object

    __format__(), __hash__(), __reduce_ex__(), __sizeof__(), __subclasshook__()

5.7.2 Properties

Name	Description
Inherited from exceptions.BaseException	
args, message	
Inherited from object	
__class__	

6 Module posixpath

Common operations on Posix pathnames.

Instead of importing this module directly, import os and refer to this module as os.path. The "os.path" name is an alias for this module on Posix systems; on other systems (e.g. Mac, Windows), os.path provides the same operations in a manner specific to that platform, and is an alias to another module (e.g. macpath, ntpath).

Some of this can actually be useful on non-Posix systems too, e.g. for manipulation of the pathname component of URLs.

6.1 Functions

abspath(*path*)
Return an absolute path.

basename(*p*)
Returns the final component of a pathname

commonprefix(*m*)
Given a list of pathnames, returns the longest common leading component

dirname(*p*)
Returns the directory component of a pathname

exists(*path*)
Test whether a path exists. Returns False for broken symbolic links

expanduser(*path*)
Expand ~ and ~user constructions. If user or $HOME is unknown, do nothing.

expandvars(*path*)
Expand shell variables of form $var and ${var}. Unknown variables are left unchanged.

getatime(*filename*)
Return the last access time of a file, reported by os.stat().

---

getctime(*filename*)

Return the metadata change time of a file, reported by os.stat().

---

getmtime(*filename*)

Return the last modification time of a file, reported by os.stat().

---

getsize(*filename*)

Return the size of a file, reported by os.stat().

---

isabs(*s*)

Test whether a path is absolute

---

isdir(*s*)

Return true if the pathname refers to an existing directory.

---

isfile(*path*)

Test whether a path is a regular file

---

islink(*path*)

Test whether a path is a symbolic link

---

ismount(*path*)

Test whether a path is a mount point

---

join(*a*, **p*)

Join two or more pathname components, inserting '/' as needed. If any component is an absolute path, all previous path components will be discarded. An empty last part will result in a path that ends with a separator.

---

lexists(*path*)

Test whether a path exists. Returns True for broken symbolic links

---

normcase(*s*)

Normalize case of pathname. Has no effect under Posix

---

normpath(*path*)

Normalize path, eliminating double slashes, etc.

---

realpath(*filename*)

Return the canonical path of the specified filename, eliminating any symbolic links encountered in the path.

---

relpath(*path*, *start='.'*)

Return a relative version of a path

---

samefile(*f1*, *f2*)

Test whether two pathnames reference the same actual file

---

sameopenfile(*fp1*, *fp2*)

Test whether two open file objects reference the same file

---

samestat(*s1*, *s2*)

Test whether two stat buffers reference the same file

---

split(*p*)

Split a pathname. Returns tuple "(head, tail)" where "tail" is everything after the final slash. Either part may be empty.

---

splitdrive(*p*)

Split a pathname into drive and path. On Posix, drive is always empty.

---

splitext(*p*)

Split the extension from a pathname.

Extension is everything from the last dot to the end, ignoring leading dots. Returns "(root, ext)"; ext may be empty.

---

walk(*top, func, arg*)

Directory tree walk with callback function.

For each directory in the directory tree rooted at top (including top itself, but excluding '.' and '..'), call func(arg, dirname, fnames). dirname is the name of the directory, and fnames a list of the names of the files and subdirectories in dirname (excluding '.' and '..'). func may modify the fnames list in-place (e.g. via del or slice assignment), and walk will only recurse into the subdirectories whose names remain in fnames; this can be used to implement a filter, or to impose a specific order of visiting. No semantics are defined for, or required of, arg, beyond that arg is always passed to func. It can be used, e.g., to pass a filename pattern, or a mutable object designed to accumulate statistics. Passing None for arg is common.

6.2 Variables

Name	Description
altsep	**Value:** `None`
curdir	**Value:** `'.'`
defpath	**Value:** `':/bin:/usr/bin'`
devnull	**Value:** `'/dev/null'`
extsep	**Value:** `'.'`
pardir	**Value:** `'..'`
pathsep	**Value:** `':'`
sep	**Value:** `'/'`
supports_unicode_filename-s	**Value:** `True`

7 Package pygame

Pygame is a set of Python modules designed for writing games. It is written on top of the excellent SDL library. This allows you to create fully featured games and multimedia programs in the python language. The package is highly portable, with games running on Windows, MacOS, OS X, BeOS, FreeBSD, IRIX, and Linux.

Version: 1.9.2a0

7.1 Functions

packager_imports()
some additional imports that py2app/py2exe will want to see

warn_unwanted_files()
warn about unneeded old files

7.2 Variables

Name	Description
ACTIVEEVENT	**Value:** 1
ANYFORMAT	**Value:** 268435456
ASYNCBLIT	**Value:** 4
AUDIO_S16	**Value:** 32784
AUDIO_S16LSB	**Value:** 32784
AUDIO_S16MSB	**Value:** 36880
AUDIO_S16SYS	**Value:** 32784
AUDIO_S8	**Value:** 32776
AUDIO_U16	**Value:** 16
AUDIO_U16LSB	**Value:** 16
AUDIO_U16MSB	**Value:** 4112
AUDIO_U16SYS	**Value:** 16
AUDIO_U8	**Value:** 8
BIG_ENDIAN	**Value:** 4321
BLEND_ADD	**Value:** 1
BLEND_MAX	**Value:** 5
BLEND_MIN	**Value:** 4
BLEND_MULT	**Value:** 3
BLEND_PREMULTIPLI-ED	**Value:** 17
BLEND_RGBA_ADD	**Value:** 6

Name	Description
BLEND_RGBA_MAX	**Value:** 16
BLEND_RGBA_MIN	**Value:** 9
BLEND_RGBA_MULT	**Value:** 8
BLEND_RGBA_SUB	**Value:** 7
BLEND_RGB_ADD	**Value:** 1
BLEND_RGB_MAX	**Value:** 5
BLEND_RGB_MIN	**Value:** 4
BLEND_RGB_MULT	**Value:** 3
BLEND_RGB_SUB	**Value:** 2
BLEND_SUB	**Value:** 2
BUTTON_X1	**Value:** 6
BUTTON_X2	**Value:** 7
DOUBLEBUF	**Value:** 1073741824
FULLSCREEN	**Value:** -2147483648
GL_ACCELERATED_VIS-UAL	**Value:** 15
GL_ACCUM_ALPHA_SIZ-E	**Value:** 11
GL_ACCUM_BLUE_SIZE	**Value:** 10
GL_ACCUM_GREEN_SIZ-E	**Value:** 9
GL_ACCUM_RED_SIZE	**Value:** 8
GL_ALPHA_SIZE	**Value:** 3
GL_BLUE_SIZE	**Value:** 2
GL_BUFFER_SIZE	**Value:** 4
GL_DEPTH_SIZE	**Value:** 6
GL_DOUBLEBUFFER	**Value:** 5
GL_GREEN_SIZE	**Value:** 1
GL_MULTISAMPLEBUF-FERS	**Value:** 13
GL_MULTISAMPLESAM-PLES	**Value:** 14
GL_RED_SIZE	**Value:** 0
GL_STENCIL_SIZE	**Value:** 7
GL_STEREO	**Value:** 12
GL_SWAP_CONTROL	**Value:** 16
HAT_CENTERED	**Value:** 0
HAT_DOWN	**Value:** 4
HAT_LEFT	**Value:** 8
HAT_LEFTDOWN	**Value:** 12
HAT_LEFTUP	**Value:** 9
HAT_RIGHT	**Value:** 2

Name	Description
HAT_RIGHTDOWN	**Value:** 6
HAT_RIGHTUP	**Value:** 3
HAT_UP	**Value:** 1
HAVE_NEWBUF	**Value:** 1
HWACCEL	**Value:** 256
HWPALETTE	**Value:** 536870912
HWSURFACE	**Value:** 1
IYUV_OVERLAY	**Value:** 1448433993
JOYAXISMOTION	**Value:** 7
JOYBALLMOTION	**Value:** 8
JOYBUTTONDOWN	**Value:** 10
JOYBUTTONUP	**Value:** 11
JOYHATMOTION	**Value:** 9
KEYDOWN	**Value:** 2
KEYUP	**Value:** 3
KMOD_ALT	**Value:** 768
KMOD_CAPS	**Value:** 8192
KMOD_CTRL	**Value:** 192
KMOD_LALT	**Value:** 256
KMOD_LCTRL	**Value:** 64
KMOD_LMETA	**Value:** 1024
KMOD_LSHIFT	**Value:** 1
KMOD_META	**Value:** 3072
KMOD_MODE	**Value:** 16384
KMOD_NONE	**Value:** 0
KMOD_NUM	**Value:** 4096
KMOD_RALT	**Value:** 512
KMOD_RCTRL	**Value:** 128
KMOD_RMETA	**Value:** 2048
KMOD_RSHIFT	**Value:** 2
KMOD_SHIFT	**Value:** 3
K_0	**Value:** 48
K_1	**Value:** 49
K_2	**Value:** 50
K_3	**Value:** 51
K_4	**Value:** 52
K_5	**Value:** 53
K_6	**Value:** 54
K_7	**Value:** 55
K_8	**Value:** 56
K_9	**Value:** 57
K_AMPERSAND	**Value:** 38

Name	Description
K_ASTERISK	**Value:** 42
K_AT	**Value:** 64
K_BACKQUOTE	**Value:** 96
K_BACKSLASH	**Value:** 92
K_BACKSPACE	**Value:** 8
K_BREAK	**Value:** 318
K_CAPSLOCK	**Value:** 301
K_CARET	**Value:** 94
K_CLEAR	**Value:** 12
K_COLON	**Value:** 58
K_COMMA	**Value:** 44
K_DELETE	**Value:** 127
K_DOLLAR	**Value:** 36
K_DOWN	**Value:** 274
K_END	**Value:** 279
K_EQUALS	**Value:** 61
K_ESCAPE	**Value:** 27
K_EURO	**Value:** 321
K_EXCLAIM	**Value:** 33
K_F1	**Value:** 282
K_F10	**Value:** 291
K_F11	**Value:** 292
K_F12	**Value:** 293
K_F13	**Value:** 294
K_F14	**Value:** 295
K_F15	**Value:** 296
K_F2	**Value:** 283
K_F3	**Value:** 284
K_F4	**Value:** 285
K_F5	**Value:** 286
K_F6	**Value:** 287
K_F7	**Value:** 288
K_F8	**Value:** 289
K_F9	**Value:** 290
K_FIRST	**Value:** 0
K_GREATER	**Value:** 62
K_HASH	**Value:** 35
K_HELP	**Value:** 315
K_HOME	**Value:** 278
K_INSERT	**Value:** 277
K_KP0	**Value:** 256
K_KP1	**Value:** 257

Name	Description
K_KP2	**Value:** 258
K_KP3	**Value:** 259
K_KP4	**Value:** 260
K_KP5	**Value:** 261
K_KP6	**Value:** 262
K_KP7	**Value:** 263
K_KP8	**Value:** 264
K_KP9	**Value:** 265
K_KP_DIVIDE	**Value:** 267
K_KP_ENTER	**Value:** 271
K_KP_EQUALS	**Value:** 272
K_KP_MINUS	**Value:** 269
K_KP_MULTIPLY	**Value:** 268
K_KP_PERIOD	**Value:** 266
K_KP_PLUS	**Value:** 270
K_LALT	**Value:** 308
K_LAST	**Value:** 323
K_LCTRL	**Value:** 306
K_LEFT	**Value:** 276
K_LEFTBRACKET	**Value:** 91
K_LEFTPAREN	**Value:** 40
K_LESS	**Value:** 60
K_LMETA	**Value:** 310
K_LSHIFT	**Value:** 304
K_LSUPER	**Value:** 311
K_MENU	**Value:** 319
K_MINUS	**Value:** 45
K_MODE	**Value:** 313
K_NUMLOCK	**Value:** 300
K_PAGEDOWN	**Value:** 281
K_PAGEUP	**Value:** 280
K_PAUSE	**Value:** 19
K_PERIOD	**Value:** 46
K_PLUS	**Value:** 43
K_POWER	**Value:** 320
K_PRINT	**Value:** 316
K_QUESTION	**Value:** 63
K_QUOTE	**Value:** 39
K_QUOTEDBL	**Value:** 34
K_RALT	**Value:** 307
K_RCTRL	**Value:** 305
K_RETURN	**Value:** 13

Name	Description
K_RIGHT	**Value:** 275
K_RIGHTBRACKET	**Value:** 93
K_RIGHTPAREN	**Value:** 41
K_RMETA	**Value:** 309
K_RSHIFT	**Value:** 303
K_RSUPER	**Value:** 312
K_SCROLLOCK	**Value:** 302
K_SEMICOLON	**Value:** 59
K_SLASH	**Value:** 47
K_SPACE	**Value:** 32
K_SYSREQ	**Value:** 317
K_TAB	**Value:** 9
K_UNDERSCORE	**Value:** 95
K_UNKNOWN	**Value:** 0
K_UP	**Value:** 273
K_a	**Value:** 97
K_b	**Value:** 98
K_c	**Value:** 99
K_d	**Value:** 100
K_e	**Value:** 101
K_f	**Value:** 102
K_g	**Value:** 103
K_h	**Value:** 104
K_i	**Value:** 105
K_j	**Value:** 106
K_k	**Value:** 107
K_l	**Value:** 108
K_m	**Value:** 109
K_n	**Value:** 110
K_o	**Value:** 111
K_p	**Value:** 112
K_q	**Value:** 113
K_r	**Value:** 114
K_s	**Value:** 115
K_t	**Value:** 116
K_u	**Value:** 117
K_v	**Value:** 118
K_w	**Value:** 119
K_x	**Value:** 120
K_y	**Value:** 121
K_z	**Value:** 122
LIL_ENDIAN	**Value:** 1234

Name	Description
MOUSEBUTTONDOWN	**Value:** 5
MOUSEBUTTONUP	**Value:** 6
MOUSEMOTION	**Value:** 4
NOEVENT	**Value:** 0
NOFRAME	**Value:** 32
NUMEVENTS	**Value:** 32
OPENGL	**Value:** 2
OPENGLBLIT	**Value:** 10
PREALLOC	**Value:** 16777216
QUIT	**Value:** 12
RESIZABLE	**Value:** 16
RLEACCEL	**Value:** 16384
RLEACCELOK	**Value:** 8192
SCRAP_BMP	**Value:** 'image/bmp'
SCRAP_CLIPBOARD	**Value:** 0
SCRAP_PBM	**Value:** 'image/pbm'
SCRAP_PPM	**Value:** 'image/ppm'
SCRAP_SELECTION	**Value:** 1
SCRAP_TEXT	**Value:** 'text/plain'
SRCALPHA	**Value:** 65536
SRCCOLORKEY	**Value:** 4096
SWSURFACE	**Value:** 0
SYSWMEVENT	**Value:** 13
TIMER_RESOLUTION	**Value:** 10
USEREVENT	**Value:** 24
USEREVENT_DROPFIL-E	**Value:** 4096
UYVY_OVERLAY	**Value:** 1498831189
VIDEOEXPOSE	**Value:** 17
VIDEORESIZE	**Value:** 16
YUY2_OVERLAY	**Value:** 844715353
YV12_OVERLAY	**Value:** 842094169
YVYU_OVERLAY	**Value:** 1431918169
__package__	**Value:** 'pygame'
movie	**Value:** ??
rev	**Value:** 'cdb77d20ee8e'
ver	**Value:** '1.9.2a0'
vernum	**Value:** (1, 9, 2)

7.3   Class BufferError

object ─┐

exceptions.BaseException ─┐

      exceptions.Exception ─┐

           exceptions.StandardError ─┐

                exceptions.BufferError ─┐

                        pygame.BufferError
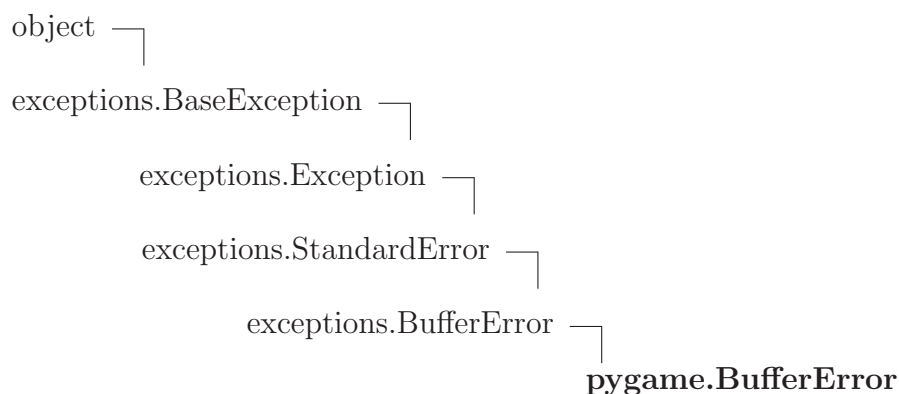
7.3.1   Methods

Inherited from exceptions.BufferError

    __init__(), __new__()

Inherited from exceptions.BaseException

    __delattr__(), __getattribute__(), __getitem__(), __getslice__(), __reduce__(), __repr__(),
    __setattr__(), __setstate__(), __str__(), __unicode__()

Inherited from object

    __format__(), __hash__(), __reduce_ex__(), __sizeof__(), __subclasshook__()

7.3.2   Properties

Name	Description
Inherited from exceptions.BaseException	
args, message	
Inherited from object	
__class__	

7.4   Class BufferProxy

object ─┐

      pygame.BufferProxy

BufferProxy(<parent>) -> BufferProxy pygame object to export a surface buffer through an array protocol

7.4.1 Methods

__new__(*T, S, ...*)

Return Value
> a new object with type S, a subtype of T

Overrides: object.__new__

__repr__(*x*)

repr(x)

Overrides: object.__repr__

---

write(*buffer, offset=0*)

Write raw bytes to object buffer.

Inherited from object

> __delattr__(), __format__(), __getattribute__(), __hash__(), __init__(), __reduce__(), __reduce_ex__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

7.4.2 Properties

Name	Description
__array_interface__	Version 3 array interface, Python level
__array_struct__	Version 3 array interface, C level
length	length -> int The size, in bytes, of the exported buffer.
parent	parent -> Surface parent -> <parent> Return wrapped exporting object.
raw	raw -> bytes A copy of the exported buffer as a single block of bytes.
Inherited from object __class__	

7.5 Class Color

object ┐
    pygame.Color

Color(name) -> Color Color(r, g, b, a) -> Color Color(rgbvalue) -> Color pygame object for color representations

7.5.1 Methods

__add__(x, y)
x+y

__coerce__(x, y)
coerce(x, y)

__delitem__(x, y)
del x[y]

__div__(x, y)
x/y

__eq__(x, y)
x==y

__float__(x)
float(x)

__floordiv__(x, y)
x//y

__ge__(x, y)
x>=y

__getitem__(x, y)
x[y]

__getslice__(*x, i, j*)

x[i:j]

Use of negative indices is not supported.

__gt__(*x, y*)

x>y

__hex__(*x*)

hex(x)

__index__(...)

x[y:z] <==> x[y.__index__():z.__index__()]

__int__(*x*)

int(x)

__invert__(*x*)

˜x

__le__(*x, y*)

x<=y

__len__(*x*)

len(x)

__long__(*x*)

long(x)

__lt__(*x, y*)

x<y

__mod__(*x, y*)

x%y

__mul__(*x, y*)

x*y

__ne__(*x, y*)

x!=y

__new__(*T, S, ...*)

Return Value

    a new object with type S, a subtype of T

Overrides: object.__new__

__oct__(*x*)

oct(x)

__radd__(*x, y*)

y+x

__rdiv__(*x, y*)

y/x

__repr__(*x*)

repr(x)

Overrides: object.__repr__

__rfloordiv__(*x, y*)

y//x

__rmod__(*x, y*)

y%x

__rmul__(*x, y*)

y*x

__rsub__(*x, y*)

y-x

__**setitem**__(*x, i, y*)

x[i]=y

__**sub**__(*x, y*)

x-y

---

correct_gamma(*...*)

correct_gamma (gamma) -> Color Applies a certain gamma value to the Color.

---

normalize()

Returns the normalized RGBA values of the Color.

Return Value
    tuple

---

set_length(*len*)

Set the number of elements in the Color to 1,2,3, or 4.

Return Value
    None

Inherited from object

__delattr__(), __format__(), __getattribute__(), __hash__(), __init__(), __reduce__(), __reduce_ex__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

7.5.2 Properties

Name	Description
__array_struct__	array structure interface, read only
a	a -> int Gets or sets the alpha value of the Color.
b	b -> int Gets or sets the blue value of the Color.
cmy	cmy -> tuple Gets or sets the CMY representation of the Color.
g	g -> int Gets or sets the green value of the Color.
hsla	hsla -> tuple Gets or sets the HSLA representation of the Color.
hsva	hsva -> tuple Gets or sets the HSVA representation of the Color.

Name	Description
i1i2i3	i1i2i3 -> tuple Gets or sets the I1I2I3 representation of the Color.
r	r -> int Gets or sets the red value of the Color.
Inherited from object	
__class__	

7.6 Class Overlay

object ┐
       pygame.Overlay

Overlay(format, (width, height)) -> Overlay pygame object for video overlay graphics

7.6.1 Methods

__new__(*T, S, ...*)

Return Value
     a new object with type S, a subtype of T

Overrides: object.__new__

---

display(*...*)

display((y, u, v)) -> None display() -> None set the overlay pixel data

---

get_hardware(*rect*)

test if the Overlay is hardware accelerated

Return Value
     int

---

set_location(*rect*)

control where the overlay is displayed

Return Value
     None

Inherited from object

     __delattr__(), __format__(), __getattribute__(), __hash__(), __init__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

7.6.2   Properties

Name	Description
Inherited from object	
__class__	

7.7   Class PixelArray

object ─┐
        pygame.PixelArray

PixelArray(Surface) -> PixelArray pygame object for direct pixel access of surfaces

7.7.1   Methods

__contains__(*x, y*)
y in x

__delitem__(*x, y*)
del x[y]

__getitem__(*x, y*)
x[y]

__iter__(*x*)
iter(x)

__len__(*x*)
len(x)

__new__(*T, S, ...*)
Return Value
a new object with type S, a subtype of T
Overrides: object.__new__

__repr__(*x*)

repr(x)

Overrides: object.__repr__

__setitem__(*x, i, y*)

x[i]=y

---

compare(*array, distance*=0, *weights*=(0.299, 0.587, 0.114))

Compares the PixelArray with another one.

Return Value
    PixelArray

---

extract(*color, distance*=0, *weights*=(0.299, 0.587, 0.114))

Extracts the passed color from the PixelArray.

Return Value
    PixelArray

---

make_surface()

Creates a new Surface from the current PixelArray.

Return Value
    Surface

---

replace(*color, repcolor, distance*=0, *weights*=(0.299, 0.587, 0.114))

Replaces the passed color in the PixelArray with another one.

Return Value
    None

---

transpose()

Exchanges the x and y axis.

Return Value
    PixelArray

Inherited from object

__delattr__(), __format__(), __getattribute__(), __hash__(), __init__(), __reduce__(), __reduce_ex__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

7.7.2 Properties

Name	Description
__array_interface__	Version 3
__array_struct__	Version 3
itemsize	itemsize -> int Returns the byte size of a pixel array item
ndim	ndim -> int Returns the number of dimensions.
shape	shape -> tuple of int's Returns the array size.
strides	strides -> tuple of int's Returns byte offsets for each array dimension.
surface	surface -> Surface Gets the Surface the PixelArray uses.
Inherited from object __class__	

7.8 Class Rect

object ─┐
    │
    pygame.Rect

Rect(left, top, width, height) -> Rect Rect((left, top), (width, height)) -> Rect Rect(object) -> Rect pygame object for storing rectangular coordinates

7.8.1 Methods

__coerce__(*x, y*)

coerce(x, y)

__copy__(...)

__delitem__(*x, y*)

del x[y]

__delslice__(*x, i, j*)

del x[i:j]

Use of negative indices is not supported.

__eq__(*x, y*)

x==y

__ge__(*x, y*)

x>=y

__getitem__(*x, y*)

x[y]

__getslice__(*x, i, j*)

x[i:j]

Use of negative indices is not supported.

__gt__(*x, y*)

x>y

__init__(*left, top, width, height*)

x.__init__(...) initializes x; see help(type(x)) for signature

Return Value
    Rect

Overrides: object.__init__

__le__(*x, y*)

x<=y

__len__(*x*)

len(x)

__lt__(*x, y*)

x<y

__ne__(*x, y*)

x!=y

__new__(*T, S, ...*)

Return Value
    a new object with type S, a subtype of T

Overrides: object.__new__

__nonzero__(*x*)

x != 0

__reduce__(*...*)

helper for pickle

Overrides: object.__reduce__ extit(inherited documentation)

__repr__(*x*)

repr(x)

Overrides: object.__repr__

__setitem__(*x, i, y*)

x[i]=y

__setslice__(*x, i, j, y*)

x[i:j]=y

Use of negative indices is not supported.

__str__(*x*)

str(x)

Overrides: object.__str__

---

clamp(*Rect*)

moves the rectangle inside another

Return Value
    Rect

---

clamp_ip(*Rect*)

moves the rectangle inside another, in place

Return Value
    None

---

clip(*Rect*)

crops a rectangle inside another

Return Value
    Rect

---

collidedict(*dict*)

test if one rectangle in a dictionary intersects

Return Value
    (key, value)

---

collidedictall(*dict*)

test if all rectangles in a dictionary intersect

Return Value
    [(key, value), ...]

---

collidelist(*list*)

test if one rectangle in a list intersects

Return Value
    index

---

collidelistall(*list*)

test if all rectangles in a list intersect

Return Value
    indices

---

collidepoint(*x, y*)

collidepoint((x,y)) -> bool test if a point is inside a rectangle

Return Value
    bool

colliderect(*Rect*)

test if two rectangles overlap

Return Value
    bool

---

contains(*Rect*)

test if one rectangle is inside another

Return Value
    bool

---

copy()

copy the rectangle

Return Value
    Rect

---

fit(*Rect*)

resize and move a rectangle with aspect ratio

Return Value
    Rect

---

inflate(*x*, *y*)

grow or shrink the rectangle size

Return Value
    Rect

---

inflate_ip(*x*, *y*)

grow or shrink the rectangle size, in place

Return Value
    None

---

move(*x*, *y*)

moves the rectangle

Return Value
    Rect

---

move_ip(*x, y*)

moves the rectangle, in place

Return Value
    None

---

normalize()

correct negative sizes

Return Value
    None

---

union(*Rect*)

joins two rectangles into one

Return Value
    Rect

---

union_ip(*Rect*)

joins two rectangles into one, in place

Return Value
    None

---

unionall(*Rect_sequence*)

the union of many rectangles

Return Value
    Rect

---

unionall_ip(*Rect_sequence*)

the union of many rectangles, in place

Return Value
    None

Inherited from object

__delattr__(), __format__(), __getattribute__(), __hash__(), __reduce_ex__(), __setattr__(), __sizeof__(), __subclasshook__()

7.8.2 Properties

Name	Description
__safe_for_unpickling__	

Name	Description
bottom	
bottomleft	
bottomright	
center	
centerx	
centery	
h	
height	
left	
midbottom	
midleft	
midright	
midtop	
right	
size	
top	
topleft	
topright	
w	
width	
x	
y	
Inherited from object	
__class__	

7.9   Class Surface

object ─┐
       pygame.Surface

Surface((width, height), flags=0, depth=0, masks=None) -> Surface Surface((width, height), flags=0, Surface) -> Surface pygame object for representing images

7.9.1   Methods

__copy__()
create a new copy of a Surface
Return Value
Surface

__init__(...)

x.__init__(...) initializes x; see help(type(x)) for signature

Overrides: object.__init__

__new__(*T, S, ...*)

Return Value
    a new object with type S, a subtype of T

Overrides: object.__new__

__repr__(*x*)

repr(x)

Overrides: object.__repr__

---

blit(*source, dest, area*=None, *special_flags*=0)

draw one image onto another

Return Value
    Rect

---

convert(*Surface*)

convert(depth, flags=0) -> Surface convert(masks, flags=0) -> Surface
convert() -> Surface change the pixel format of an image

Return Value
    Surface

---

convert_alpha(*Surface*)

convert_alpha() -> Surface change the pixel format of an image including per
pixel alphas

Return Value
    Surface

---

copy()

create a new copy of a Surface

Return Value
    Surface

---

fill(*color*, *rect=*None, *special_flags=*0)

fill Surface with a solid color

Return Value
    Rect

---

get_abs_offset()

find the absolute position of a child subsurface inside its top level parent

Return Value
    (x, y)

---

get_abs_parent()

find the top level parent of a subsurface

Return Value
    Surface

---

get_alpha()

get the current Surface transparency value

Return Value
    int_value or None

---

get_at(...)

get_at((x, y)) -> Color get the color value at a single pixel

---

get_at_mapped(...)

get_at_mapped((x, y)) -> Color get the mapped color value at a single pixel

---

get_bitsize()

get the bit depth of the Surface pixel format

Return Value
    int

---

get_bounding_rect(*min_alpha=*1)

find the smallest rect containing data

Return Value
    Rect

get_buffer()

acquires a buffer object for the pixels of the Surface.

Return Value
    BufferProxy

get_bytesize()

get the bytes used per Surface pixel

Return Value
    int

get_clip()

get the current clipping area of the Surface

Return Value
    Rect

get_colorkey()

Get the current transparent colorkey

Return Value
    RGB or None

get_flags()

get the additional flags used for the Surface

Return Value
    int

get_height()

get the height of the Surface

Return Value
    height

get_locked()

test if the Surface is current locked

Return Value
    bool

---

get_locks()

Gets the locks for the Surface

Return Value
    tuple

---

get_losses()

the significant bits used to convert between a color and a mapped integer

Return Value
    (R, G, B, A)

---

get_masks()

the bitmasks needed to convert between a color and a mapped integer

Return Value
    (R, G, B, A)

---

get_offset()

find the position of a child subsurface inside a parent

Return Value
    (x, y)

---

get_palette()

get the color index palette for an 8bit Surface

Return Value
    [RGB, RGB, RGB, ...]

---

get_palette_at(*index*)

get the color for a single entry in a palette

Return Value
    RGB

---

get_parent()

find the parent of a subsurface

Return Value
    Surface

---

get_pitch()

get the number of bytes used per Surface row

Return Value
    int

---

get_rect(***kwargs*)

get the rectangular area of the Surface

Return Value
    Rect

---

get_shifts()

the bit shifts needed to convert between a color and a mapped integer

Return Value
    (R, G, B, A)

---

get_size()

get the dimensions of the Surface

Return Value
    (width, height)

---

get_view(*...*)

get_view(<kind>='2') -> BufferProxy return a buffer view of the Surface's pixels.

---

get_width()

get the width of the Surface

Return Value
    width

---

lock()

lock the Surface memory for pixel access

Return Value
    None

---

map_rgb(*Color*)

convert a color into a mapped color value

Return Value
    mapped_int

---

mustlock()

test if the Surface requires locking

Return Value
    bool

---

scroll(*dx=0, dy=0*)

Shift the surface image in place

Return Value
    None

---

set_alpha(*value, flags=0*)

set_alpha(None) -> None set the alpha value for the full Surface image

Return Value
    None

---

set_at(*...*)

set_at((x, y), Color) -> None set the color value for a single pixel

---

set_clip(*rect*)

set_clip(None) -> None set the current clipping area of the Surface

Return Value
    None

---

set_colorkey(*Color, flags=0*)

set_colorkey(None) -> None Set the transparent colorkey

Return Value
    None

---

set_masks(*...*)

set_masks((r,g,b,a)) -> None set the bitmasks needed to convert between a color and a mapped integer

set_palette(*RGB=..., RGB=..., RGB=..., ...*)

set the color palette for an 8bit Surface

Return Value
    None

---

set_palette_at(*index, RGB*)

set the color for a single index in an 8bit Surface palette

Return Value
    None

---

set_shifts(*...*)

set_shifts((r,g,b,a)) -> None sets the bit shifts needed to convert between a color and a mapped integer

---

subsurface(*Rect*)

create a new surface that references its parent

Return Value
    Surface

---

unlock()

unlock the Surface memory from pixel access

Return Value
    None

---

unmap_rgb(*mapped_int*)

convert a mapped integer color value into a Color

Return Value
    Color

Inherited from object

__delattr__(), __format__(), __getattribute__(), __hash__(), __reduce__(), __reduce_ex__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

7.9.2 Properties

Name	Description
Inherited from object	
__class__	

7.10   Class Surface

object ┐

       pygame.Surface

Surface((width, height), flags=0, depth=0, masks=None) -> Surface Surface((width, height), flags=0, Surface) -> Surface pygame object for representing images

7.10.1   Methods

__copy__()

create a new copy of a Surface

Return Value
     Surface

__init__(...)

x.__init__(...) initializes x; see help(type(x)) for signature

Overrides: object.__init__

__new__(*T*, *S*, ...)

Return Value
     a new object with type S, a subtype of T

Overrides: object.__new__

__repr__(*x*)

repr(x)

Overrides: object.__repr__

---

blit(*source*, *dest*, *area*=None, *special_flags*=0)

draw one image onto another

Return Value
     Rect

---

convert(*Surface*)

convert(depth, flags=0) -> Surface convert(masks, flags=0) -> Surface convert() -> Surface change the pixel format of an image

Return Value
    Surface

convert_alpha(*Surface*)

convert_alpha() -> Surface change the pixel format of an image including per pixel alphas

Return Value
    Surface

copy()

create a new copy of a Surface

Return Value
    Surface

fill(*color*, *rect=*`None`, *special_flags=*`0`)

fill Surface with a solid color

Return Value
    Rect

get_abs_offset()

find the absolute position of a child subsurface inside its top level parent

Return Value
    (x, y)

get_abs_parent()

find the top level parent of a subsurface

Return Value
    Surface

get_alpha()

get the current Surface transparency value

Return Value
    int_value or None

---

get_at(*...*)

get_at((x, y)) -> Color get the color value at a single pixel

---

get_at_mapped(*...*)

get_at_mapped((x, y)) -> Color get the mapped color value at a single pixel

---

get_bitsize()

get the bit depth of the Surface pixel format

Return Value
    int

---

get_bounding_rect(*min_alpha=1*)

find the smallest rect containing data

Return Value
    Rect

---

get_buffer()

acquires a buffer object for the pixels of the Surface.

Return Value
    BufferProxy

---

get_bytesize()

get the bytes used per Surface pixel

Return Value
    int

---

get_clip()

get the current clipping area of the Surface

Return Value
    Rect

---

get_colorkey()

Get the current transparent colorkey

Return Value
    RGB or None

---

get_flags()

get the additional flags used for the Surface

Return Value
    int

---

get_height()

get the height of the Surface

Return Value
    height

---

get_locked()

test if the Surface is current locked

Return Value
    bool

---

get_locks()

Gets the locks for the Surface

Return Value
    tuple

---

get_losses()

the significant bits used to convert between a color and a mapped integer

Return Value
    (R, G, B, A)

---

get_masks()

the bitmasks needed to convert between a color and a mapped integer

Return Value
    (R, G, B, A)

---

get_offset()

find the position of a child subsurface inside a parent

Return Value
    (x, y)

---

get_palette()

get the color index palette for an 8bit Surface

Return Value
    [RGB, RGB, RGB, ...]

---

get_palette_at(*index*)

get the color for a single entry in a palette

Return Value
    RGB

---

get_parent()

find the parent of a subsurface

Return Value
    Surface

---

get_pitch()

get the number of bytes used per Surface row

Return Value
    int

---

get_rect(***kwargs*)

get the rectangular area of the Surface

Return Value
    Rect

---

get_shifts()

the bit shifts needed to convert between a color and a mapped integer

Return Value
    (R, G, B, A)

---

get_size()

get the dimensions of the Surface

Return Value
    (width, height)

get_view(*...*)

get_view(<kind>=’2’) -> BufferProxy return a buffer view of the Surface’s pixels.

---

get_width()

get the width of the Surface

Return Value
　　width

---

lock()

lock the Surface memory for pixel access

Return Value
　　None

---

map_rgb(*Color*)

convert a color into a mapped color value

Return Value
　　mapped_int

---

mustlock()

test if the Surface requires locking

Return Value
　　bool

---

scroll(*dx=0, dy=0*)

Shift the surface image in place

Return Value
　　None

---

set_alpha(*value, flags=0*)

set_alpha(None) -> None set the alpha value for the full Surface image

Return Value
　　None

---

set_at(*...*)

set_at((x, y), Color) -> None set the color value for a single pixel

set_clip(*rect*)

set_clip(None) -> None set the current clipping area of the Surface

Return Value
    None

---

set_colorkey(*Color, flags=*`0`)

set_colorkey(None) -> None Set the transparent colorkey

Return Value
    None

---

set_masks(*...*)

set_masks((r,g,b,a)) -> None set the bitmasks needed to convert between a color and a mapped integer

---

set_palette(*RGB=..., RGB=..., RGB=..., ...*)

set the color palette for an 8bit Surface

Return Value
    None

---

set_palette_at(*index, RGB*)

set the color for a single index in an 8bit Surface palette

Return Value
    None

---

set_shifts(*...*)

set_shifts((r,g,b,a)) -> None sets the bit shifts needed to convert between a color and a mapped integer

---

subsurface(*Rect*)

create a new surface that references its parent

Return Value
    Surface

---

unlock()

unlock the Surface memory from pixel access

Return Value
    None

unmap_rgb(*mapped_int*)
convert a mapped integer color value into a Color
Return Value   Color

Inherited from object

__delattr__(), __format__(), __getattribute__(), __hash__(), __reduce__(), __reduce_ex__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

7.10.2   Properties

Name	Description
Inherited from object	
__class__	

7.11   Class error

object ⌐

exceptions.BaseException ⌐

exceptions.Exception ⌐

exceptions.StandardError ⌐

exceptions.RuntimeError ⌐

pygame.error
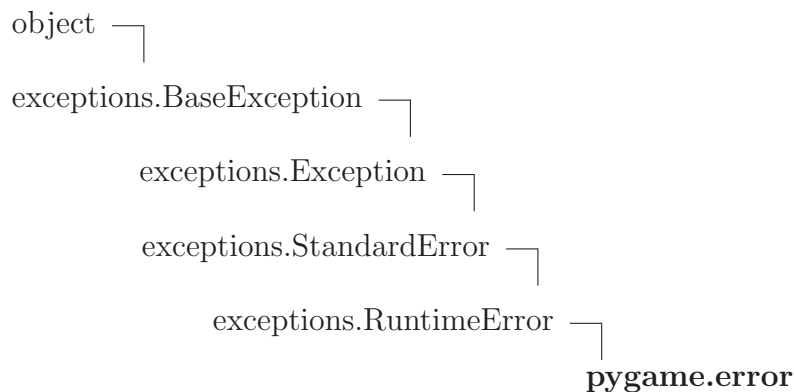
7.11.1   Methods

Inherited from exceptions.RuntimeError

__init__(), __new__()

Inherited from exceptions.BaseException

__delattr__(), __getattribute__(), __getitem__(), __getslice__(), __reduce__(), __repr__(), __setattr__(), __setstate__(), __str__(), __unicode__()

Inherited from object

__format__(), __hash__(), __reduce_ex__(), __sizeof__(), __subclasshook__()

7.11.2 Properties

Name	Description
Inherited from exceptions.BaseException	
args, message	
Inherited from object	
__class__	

8 Package pygame-asteroids

8.1 Modules

- **asteroid** *(Section ??, p. ??)*
- **asteroid'** *(Section 9, p. 92)*
- **bullet** *(Section ??, p. ??)*
- **bullet'** *(Section 10, p. 97)*
- **color** *(Section 11, p. 100)*
- **credit** *(Section ??, p. ??)*
- **credit'** *(Section 12, p. 102)*
- **file** *(Section ??, p. ??)*
- **file'**: Manipula Arquivo.
  (Section 13, p. 103)
- **font** *(Section ??, p. ??)*
- **font'** *(Section 14, p. 105)*
- **game** *(Section ??, p. ??)*
- **game'** *(Section 15, p. 106)*
- **main** *(Section 16, p. 129)*
- **menu_helper** *(Section 17, p. 130)*
- **polygon** *(Section ??, p. ??)*
- **polygon'** *(Section 18, p. 134)*
- **resource_manager** *(Section ??, p. ??)*
- **resource_manager'** *(Section 19, p. 138)*
- **score** *(Section 20, p. 142)*
- **ship** *(Section 21, p. 143)*
- **sprite_collision** *(Section 22, p. 146)*
- **start_field** *(Section 23, p. 148)*
- **state_machine** *(Section ??, p. ??)*
- **state_machine'**: Generalização de uma máquina de estado.
  (Section 24, p. 150)
- **states** *(Section ??, p. ??)*
  - **franchise** *(Section ??, p. ??)*
  - **intro** *(Section ??, p. ??)*
  - **menu**: The splash screen of the game.
    (Section 27, p. 165)
  - **seal** *(Section ??, p. ??)*
- **states'** *(Section 29, p. 175)*
  - **franchise'**: Tela como selo de classificação do jogo.
    (Section 25, p. 152)
  - **intro'**: Tela inicial do jogo.
    (Section 26, p. 161)
  - **seal'**: Tela como selo de classificação do jogo.
    (Section 28, p. 167)

- **test_audio** *(Section ??, p. ??)*
- **test_audio'** *(Section 30, p. 176)*
- **test_rotation** *(Section ??, p. ??)*
- **test_rotation'** *(Section 31, p. 202)*
- **test_startfield** *(Section ??, p. ??)*
- **test_startfield'** *(Section 32, p. 228)*
- **test_touch_buttons** *(Section ??, p. ??)*
- **test_touch_buttons'** *(Section 33, p. 230)*
- **test_virtual_keyboard** *(Section ??, p. ??)*
- **test_virtual_keyboard'** *(Section 34, p. 253)*
- **tools** *(Section ??, p. ??)*
- **tools'**: This module contains the fundamental Control class.
  (Section 35, p. 254)
- **touch_buttons** *(Section 36, p. 260)*
- **virtual_controller** *(Section 37, p. 261)*
- **virtual_keyboard** *(Section 38, p. 262)*

8.2   Variables

Name	Description
__package__	**Value:** `None`

9 Module pygame-asteroids.asteroid'

9.1 Variables

Name	Description
__package__	**Value:** 'pygame-asteroids'

9.2 Class Asteroid

pygame-asteroids.sprite_collision.SpriteCollision ─┐

object ─┐

                                pygame.sprite.Sprite ─┤

pygame-asteroids.asteroid.Asteroid

Representa o asteroid

9.2.1 Methods

__**init**__(*self*, *location*, *size*, *angle*, *game_controller*, *from_color*=(255, 0, 255, 255), *to_color*=(255, 0, 0, 255), **groups*)

Construtor

Overrides: object.__init__

---

destroy(*self*, *angle*)

Destruir asteroids

Parameters
    angle: Ângulo

---

random_angle(*self*)

Retorna angulo aleatória

---

rotate(*self*)

---

scale(*self*, *size*)

---

update(*self, time*)
Atualizar
Parameters
`time`: Delta Time
Overrides: pygame.sprite.Sprite.update

Inherited from pygame-asteroids.sprite_collision.SpriteCollision(Section 33.12)

collision(), draw2(), get_class_name(), initrect(), rectupdate(), screencollision()

Inherited from pygame.sprite.Sprite

__repr__(), add(), add_internal(), alive(), groups(), kill(), remove(), remove_internal()

Inherited from object

__delattr__(), __format__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

9.2.2 Properties

Name	Description
Inherited from object	
__class__	

9.2.3 Class Variables

Name	Description
SECTIONS	**Value:** 2
SPEED	**Value:** 0.5

9.3 Class Color

Representa cor

9.3.1   Methods

---

hex_to_rgb(*value*)

Hexadecimal para RGB

Parameters
    `value`: Valor hexadecimal

Return Value
    Cor RGB

---

rgb_to_hex(*rgb*)

Converte cor rgb em hexadecimal

Parameters
    `rgb`: Cor rgb

Return Value
    Cor Hexadecimal

---

random_color()

Retorna cor aleatória

Return Value
    Cor (r,g,b)

---

color_replace(*surface, find_color, replace_color*)

Substitui cor da image

Parameters
    `surface`:   Imagem

    `find_color`:   Cor a procurar

    `replace_color`: Cor a substituir

Return Value
    Image Modificada

9.3.2   Class Variables

Name	Description
WHITE	Branco **Value:** (255, 255, 255)
RED	Vermelho **Value:** (255, 0, 0)

Name	Description
GREEN	Verde **Value:** (0, 255, 0)
BLACK	Preto **Value:** (0, 0, 0)
YELLOW	Amarelo **Value:** (255, 255, 0)
BLUE	Azul **Value:** (68, 204, 230)
TRANSPARENT	Transparente **Value:** (255, 255, 255, 0)
PINK	Rosa **Value:** (255, 0, 255, 255)
GRAY	Cinza **Value:** (228, 228, 228)

9.4   Class SpriteCollision

Known Subclasses: pygame-asteroids.asteroid.Asteroid, pygame-asteroids.bullet.Bullet, pygame-asteroids.ship.Ship

Classe principal que controla sprite e aplicada a todos os objetos em movimento

9.4.1   Methods

---

initrect(*self, location, game_controller*)

Iniciliza retângulo

Parameters
    location:

    game_controller:

---

get_class_name(*self*)

Retorna nome da classe instanciada

Return Value
    Nome

---

screencollision(*self, time*)

Envolve a tela do jogo para objetos

Parameters
    `time`: Delta time

---

rectupdate(*self*)

---

define qual borda colide com o objeto de colisão / blitting

---

draw2(*self, screen*)

---

blit de imagens para corrigir os lugares quando perto cantos / lados.

Parameters
    `screen`: Tela

---

collision(*self, object*)

Detecta colisão

Parameters
    `object`: Objeto

Return Value
    Booleano

10 Module pygame-asteroids.bullet'

10.1 Variables

Name	Description
__package__	**Value:** 'pygame-asteroids'

10.2 Class Bullet

pygame-asteroids.sprite_collision.SpriteCollision ——

object ——

pygame.sprite.Sprite ——

pygame-asteroids.bullet.Bullet

Bala

10.2.1 Methods

__**init**__(*self, location, angle, game_controller, *groups*)

Construtor

Parameters
    location:

    angle:

    game_controller:

    groups:

Return Value
    Instância

Overrides: object.__init__

---

destroy(*self*)

Destruir

---

update(*self, time*)

Atualizar

Parameters
    `time`: Delta time

Overrides: pygame.sprite.Sprite.update

Inherited from pygame-asteroids.sprite_collision.SpriteCollision(Section 33.12)

collision(), draw2(), get_class_name(), initrect(), rectupdate(), screencollision()

Inherited from pygame.sprite.Sprite

__repr__(), add(), add_internal(), alive(), groups(), kill(), remove(), remove_internal()

Inherited from object

__delattr__(), __format__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

10.2.2 Properties

Name	Description
Inherited from object	
__class__	

10.3 Class SpriteCollision

Known Subclasses: pygame-asteroids.asteroid.Asteroid, pygame-asteroids.bullet.Bullet, pygame-asteroids.ship.Ship

Classe principal que controla sprite e aplicada a todos os objetos em movimento

10.3.1 Methods

---

initrect(*self, location, game_controller*)

Iniciliza retângulo

Parameters
    `location`:

    `game_controller`:

---

get_class_name(*self*)

Retorna nome da classe instanciada

Return Value
    Nome

---

screencollision(*self, time*)

Envolve a tela do jogo para objetos

Parameters
    `time`: Delta time

---

rectupdate(*self*)

define qual borda colide com o objeto de colisão / blitting

---

draw2(*self, screen*)

blit de imagens para corrigir os lugares quando perto cantos / lados.

Parameters
    `screen`: Tela

---

collision(*self, object*)

Detecta colisão

Parameters
    `object`: Objeto

Return Value
    Booleano

---

11 Module pygame-asteroids.color

11.1 Variables

Name	Description
__package__	**Value: 'pygame-asteroids'**

11.2 Class Color

Representa cor

11.2.1 Methods

---

hex_to_rgb(*value*)

Hexadecimal para RGB

Parameters
    `value`: Valor hexadecimal

Return Value
    Cor RGB

---

rgb_to_hex(*rgb*)

Converte cor rgb em hexadecimal

Parameters
    `rgb`: Cor rgb

Return Value
    Cor Hexadecimal

---

random_color()

Retorna cor aleatória

Return Value
    Cor (r,g,b)

---

color_replace(*surface*, *find_color*, *replace_color*)

Substitui cor da image

Parameters
    `surface:`   Imagem

    `find_color:`   Cor a procurar

    `replace_color:` Cor a substituir

Return Value
    Image Modificada

11.2.2 Class Variables

Name	Description
WHITE	Branco **Value:** (255, 255, 255)
RED	Vermelho **Value:** (255, 0, 0)
GREEN	Verde **Value:** (0, 255, 0)
BLACK	Preto **Value:** (0, 0, 0)
YELLOW	Amarelo **Value:** (255, 255, 0)
BLUE	Azul **Value:** (68, 204, 230)
TRANSPARENT	Transparente **Value:** (255, 255, 255, 0)
PINK	Rosa **Value:** (255, 0, 255, 255)
GRAY	Cinza **Value:** (228, 228, 228)

12   Module pygame-asteroids.credit'

12.1   Functions

---

credit(*text, font, color, fps*=40, *callback*=None)

Créditos rolando verticalmente

Parameters
    text: Texto

    font: Fonte

    color: Cor

---

credit_from_file(*file_name, font, color, fps*=40, *callback*=None)

---

decode_joystick_arrow_to_keyboard_key(*joystick*)

---

is_event_type_an_input_down(*event*)

---

vc_get_angle(*center, pos*)

Parameters
    pos: posição do mouse

Return Value
    ângulo em raidanos da posição do mouse em relação ao centro informado. Controle Virtual é dividido em três partes. Para o eixo-y : 1) above the center of the controller 2) below the center of the controller 3) at the same height of the center of the controller If the mouse point is above the center of the controller, than check for one of three conditions: 1) x is to the right of the controller 2) x is to the left of the controller 3) x is at the same point as the centerx of the controller

12.2   Variables

Name	Description
__package__	**Value:** 'pygame-asteroids'

13   Module pygame-asteroids.file'

Manipula Arquivo.

Version: 1.0

Author: Humberto Lino

13.1   Variables

Name	Description
__package__	**Value:** `'pygame-asteroids'`

13.2   Class DataHandler

13.2.1   Methods

defaultscore(*self*)

13.3   Class FileManager

13.3.1   Methods

get_pickle_fixed_protocol(*self*)

---

load(*self*, *file_name*, *file_format*={`'asteroide': -1, 'controle': 1,` `'janela': 2, 'mostrar_mo...`, *sep*=`' '`)

Carregar arquivo

Parameters
    `file_name:`   Nome do arquivo

    `file_format:` Formato do arquivo

    `sep:`   Separador

Return Value
    Mapa preenchido

---

marshal(*self*, *file_name*, *object*)

unmarshal(*self, file_name*)

unmarshal_marshal(*self, file_name, default_object*)

13.3.2 Class Variables

Name	Description
CHARSET	**Value:** 'utf-8'

14 Module pygame-asteroids.font'

14.1 Variables

Name	Description
__package__	**Value:** None

14.2 Class Font

14.2.1 Class Variables

Name	Description
LARGE	**Value:** 60
MEDIUM	**Value:** 40
NORMAL	**Value:** 30
SMALL	**Value:** 20
VERY_SMALL	**Value:** 15

15 Module pygame-asteroids.game'

15.1 Functions

decode_joystick_arrow_to_keyboard_key(*joystick*)

getpagesize(*...*)

getrlimit(*...*)

getrusage(*...*)

is_event_type_an_input_down(*event*)

setrlimit(*...*)

vc_get_angle(*center, pos*)

Parameters
    pos: posição do mouse

Return Value
    ângulo em raidanos da posição do mouse em relação ao centro informado. Controle Virtual é dividido em três partes. Para o eixo-y : 1) above the center of the controller 2) below the center of the controller 3) at the same height of the center of the controller If the mouse point is above the center of the controller, than check for one of three conditions: 1) x is to the right of the controller 2) x is to the left of the controller 3) x is at the same point as the centerx of the controller

15.2 Variables

Name	Description
RLIMIT_AS	**Value:** 5
RLIMIT_CORE	**Value:** 4
RLIMIT_CPU	**Value:** 0
RLIMIT_DATA	**Value:** 2
RLIMIT_FSIZE	**Value:** 1
RLIMIT_MEMLOCK	**Value:** 6
RLIMIT_NOFILE	**Value:** 8
RLIMIT_NPROC	**Value:** 7

Name	Description
RLIMIT_RSS	**Value:** 5
RLIMIT_STACK	**Value:** 3
RLIM_INFINITY	**Value:** 9223372036854775807
RUSAGE_CHILDREN	**Value:** -1
RUSAGE_SELF	**Value:** 0
__package__	**Value:** 'pygame-asteroids'
__warningregistry__	**Value:** {('Not importing directory \'/Users/tux/git/pygame-astero...

15.3 Class Asteroid

pygame-asteroids.sprite_collision.SpriteCollision

object

pygame.sprite.Sprite

pygame-asteroids.asteroid.Asteroid
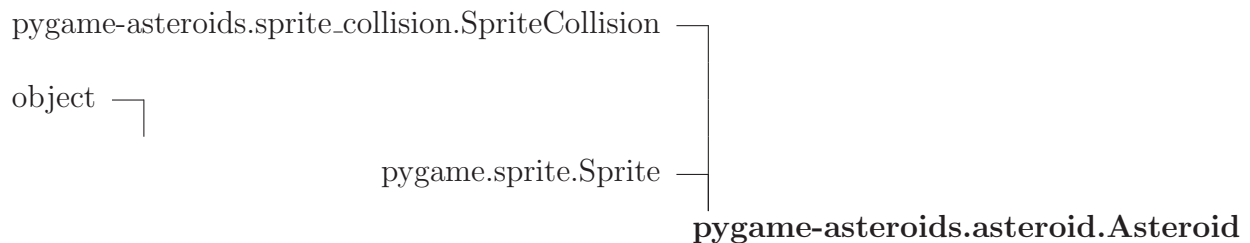
Representa o asteroid

15.3.1 Methods

__init__(*self, location, size, angle, game_controller, from_color*=(255, 0, 255, 255), *to_color*=(255, 0, 0, 255), **groups*)

Construtor

Overrides: object.__init__

---

destroy(*self, angle*)

Destruir asteroids

Parameters
    angle: Ângulo

---

random_angle(*self*)

Retorna angulo aleatória

---

rotate(*self*)

118

---

scale(*self*, *size*)

---

update(*self*, *time*)

Atualizar

Parameters
    `time`: Delta Time

Overrides: pygame.sprite.Sprite.update

Inherited from pygame-asteroids.sprite_collision.SpriteCollision(Section 33.12)

collision(), draw2(), get_class_name(), initrect(), rectupdate(), screencollision()

Inherited from pygame.sprite.Sprite

__repr__(), add(), add_internal(), alive(), groups(), kill(), remove(), remove_internal()

Inherited from object

__delattr__(), __format__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

15.3.2   Properties

Name	Description
Inherited from object	
__class__	

15.3.3   Class Variables

Name	Description
SECTIONS	**Value:** 2
SPEED	**Value:** `0.5`

15.4   Class Bullet

pygame-asteroids.sprite_collision.SpriteCollision ⎯

object ⎯

                                             pygame.sprite.Sprite ⎯

                                                    pygame-asteroids.bullet.Bullet

Bala

15.4.1   Methods

__init__(*self, location, angle, game_controller, *groups*)

Construtor

Parameters
    location:

    angle:

    game_controller:

    groups:

Return Value
    Instância

Overrides: object.__init__

---

destroy(*self*)

Destruir

---

update(*self, time*)

Atualizar

Parameters
    time: Delta time

Overrides: pygame.sprite.Sprite.update

Inherited from pygame-asteroids.sprite_collision.SpriteCollision(Section 33.12)

collision(), draw2(), get_class_name(), initrect(), rectupdate(), screencollision()

Inherited from pygame.sprite.Sprite

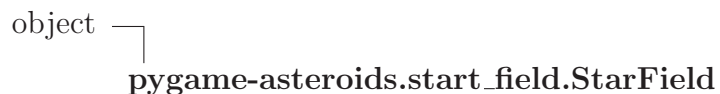__repr__(), add(), add_internal(), alive(), groups(), kill(), remove(), remove_internal()

Inherited from object

__delattr__(), __format__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

15.4.2   Properties

Name	Description
Inherited from object	
␣class␣␣	

15.5   Class Color

Representa cor

15.5.1   Methods

---

hex_to_rgb(*value*)

Hexadecimal para RGB

Parameters
    `value`: Valor hexadecimal

Return Value
    Cor RGB

---

rgb_to_hex(*rgb*)

Converte cor rgb em hexadecimal

Parameters
    `rgb`: Cor rgb

Return Value
    Cor Hexadecimal

---

random_color()

Retorna cor aleatória

Return Value
    Cor (r,g,b)

---

color_replace(*surface, find_color, replace_color*)

Substitui cor da image

Parameters
    `surface:`   Imagem

    `find_color:`   Cor a procurar

    `replace_color:` Cor a substituir

Return Value
    Image Modificada

15.5.2 Class Variables

Name	Description
WHITE	Branco **Value:** (255, 255, 255)
RED	Vermelho **Value:** (255, 0, 0)
GREEN	Verde **Value:** (0, 255, 0)
BLACK	Preto **Value:** (0, 0, 0)
YELLOW	Amarelo **Value:** (255, 255, 0)
BLUE	Azul **Value:** (68, 204, 230)
TRANSPARENT	Transparente **Value:** (255, 255, 255, 0)
PINK	Rosa **Value:** (255, 0, 255, 255)
GRAY	Cinza **Value:** (228, 228, 228)

15.6 Class DataHandler

15.6.1 Methods

---

defaultscore(*self*)

15.7   Class FileManager

15.7.1   Methods

---
get_pickle_fixed_protocol(*self*)

---
load(*self*, *file_name*, *file_format*={'asteroide': -1, 'controle': 1, 'janela': 2, 'mostrar_mo...*, *sep*=' ')

Carregar arquivo

Parameters
  file_name: Nome do arquivo

  file_format: Formato do arquivo

  sep: Separador

Return Value
  Mapa preenchido

---
marshal(*self*, *file_name*, *object*)

---
unmarshal(*self*, *file_name*)

---
unmarshal_marshal(*self*, *file_name*, *default_object*)

15.7.2   Class Variables

Name	Description
CHARSET	**Value:** 'utf-8'

15.8   Class Font

15.8.1   Class Variables

Name	Description
LARGE	**Value:** 60
MEDIUM	**Value:** 40
NORMAL	**Value:** 30
SMALL	**Value:** 20
VERY_SMALL	**Value:** 15

15.9   Class Game

15.9.1   Methods

__init__(*self, map*)

---
create_empty_actions()

15.10   Class GameController

object ─┐
           └ **pygame-asteroids.game.GameController**

Controlador do Jogo

15.10.1   Methods

__init__(*self, game_name, screen, file_manager, asset_manager, config, score_data*)

Construtor

Parameters
    `screen:`

    `file_manager:`

    `asset_manager:`

    `config:`

    `score_data:`

Overrides: object.__init__

---
announcelevel(*self, time, screen*)

Mostrar na tela o nível

Parameters
    `time:`   Delta time

    `screen:` Tela

---
create_asteroids(*self*)

Criar asteroids

---

create_random_asteroids(*self, count*)

Cria asteroids aleatoriamente

Parameters
    `self:`

Return Value
    asteroids

---

draw_game_over(*self, screen, event, time*)

Desenhar fim de jogo

Parameters
    `screen:` Tela

    `event:` Evento

    `time:`   Delta Time

---

draw_hud(*self, screen, time*)

Desenha head-up display

Parameters
    `screen:` Tela

---

get_display_center(*self*)

---

get_user_score_on_ranking(*self*)

---

initialize_joystick(*self*)

---

main(*self, screen*)

---

quit(*self*)

---

random_position(*self*)

---

save_score(*self, user_name*)

---

screen_rect(*self*)

Retangulo com o tamanho da tela

Return Value
    Retangulo

show_fps(*self*)
Display the current FPS in the window handle if fps_visible is True.

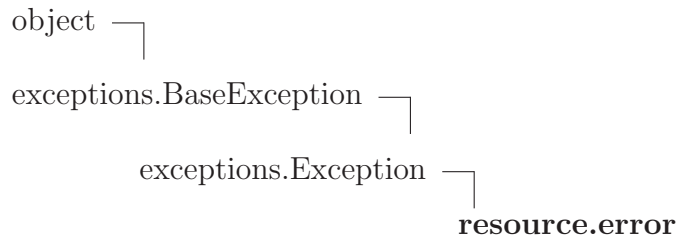## Inherited from object

__delattr__(), __format__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

15.10.2   Properties

Name	Description
Inherited from object	
__class__	

15.10.3   Class Variables

Name	Description
newhiscore	**Value:** 0
running	**Value:** 0

15.11   Class Ship

pygame-asteroids.sprite_collision.SpriteCollision ⎯

object ⎤

pygame.sprite.Sprite ⎯

pygame-asteroids.ship.Ship

Nave

15.11.1 Methods

__init__(*self, location, game_controller, *groups*)

Construtor

Parameters
    `location`:

    `game_controller`:

    `groups`:

Return Value
    Instância

Overrides: object.__init__

---

accelerate(*self, angle*)

---

accelerate_speed(*self, angle, x, y*)

Acelerar

Parameters
    `angle`: Ângulo

    `x`:   Posição X

    `y`:   Posição Y

---

shoot(*self*)

Atirar

---

get_center(*self*)

---

explode(*self, time*)

Explidir nave

Parameters
    `time`: Delta Time

---

left(*self, time*)

---

right(*self, time*)

---

up(*self*)

rotate(*self, angle*)

update(*self, time*)
Atualizar
Parameters   `time`: Delta Time
Overrides: pygame.sprite.Sprite.update

Inherited from pygame-asteroids.sprite_collision.SpriteCollision(Section 33.12)

collision(), draw2(), get_class_name(), initrect(), rectupdate(), screencollision()

Inherited from pygame.sprite.Sprite

__repr__(), add(), add_internal(), alive(), groups(), kill(), remove(), remove_internal()

Inherited from object

__delattr__(), __format__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

15.11.2 Properties

Name	Description
Inherited from object __class__	

15.11.3 Class Variables

Name	Description
SPEED_ACCELERATE	Velociadade de aceleração **Value:** `0.025`
SPEED_ROTATE	Velocidade de rotação **Value:** `0.4`
TIME_LOAD	Tempo para carregar **Value:** `200.0`
TIME_RELOAD_SHOT	Tempo para recarregar **Value:** `100`

15.12   Class SpriteCollision

Known Subclasses: pygame-asteroids.asteroid.Asteroid, pygame-asteroids.bullet.Bullet,
pygame-asteroids.ship.Ship

Classe principal que controla sprite e aplicada a todos os objetos em movimento

15.12.1   Methods

---

initrect(*self, location, game_controller*)

Iniciliza retângulo

Parameters
    `location`:

    `game_controller`:

---

get_class_name(*self*)

Retorna nome da classe instanciada

Return Value
    Nome

---

screencollision(*self, time*)

Envolve a tela do jogo para objetos

Parameters
    `time`: Delta time

---

rectupdate(*self*)

define qual borda colide com o objeto de colisão / blitting

---

draw2(*self, screen*)

blit de imagens para corrigir os lugares quando perto cantos / lados.

Parameters
    `screen`: Tela

---

collision(*self, object*)

Detecta colisão

Parameters
    `object`: Objeto

Return Value
    Booleano

15.13   Class SpriteGroup

object ⌐

pygame.sprite.AbstractGroup ⌐

            pygame.sprite.Group ⌐

                     pygame-asteroids.game.SpriteGroup
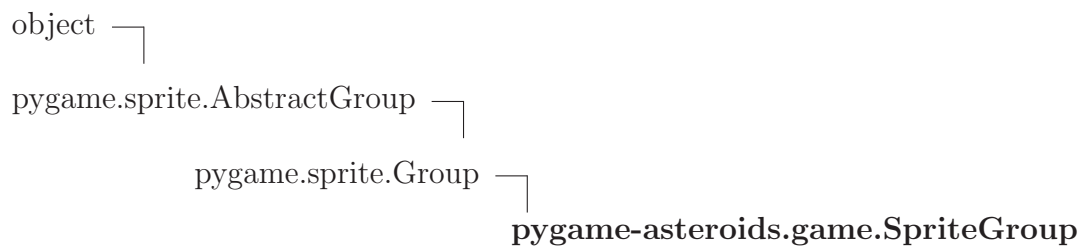
Agrupado de Sprites

15.13.1   Methods

---

draw2(*self, screen*)

Desenhar na tela

Parameters
    `screen`: Tela

Inherited from pygame.sprite.Group

    __init__()

Inherited from pygame.sprite.AbstractGroup

    __contains__(), __iter__(), __len__(), __nonzero__(), __repr__(), add(), add_internal(),
    clear(), copy(), draw(), empty(), has(), has_internal(), remove(), remove_internal(),
    sprites(), update()

Inherited from object

    __delattr__(), __format__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(),
    __setattr__(), __sizeof__(), __str__(), __subclasshook__()

15.13.2   Properties

Name	Description
Inherited from object	
__class__	

15.14   Class StarField

object ─┐
       pygame-asteroids.start_field.StarField

15.14.1   Methods

__**init**__(*self, screen*)

Create the starfield

Overrides: object.__init__

---
move(*self, start, end, direction*)

Correct for stars hitting the screen's borders

---
fix_stars_on_screen(*self, star*)

---
draw(*self*)

Inherited from object

__delattr__(), __format__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

15.14.2   Properties

Name	Description
Inherited from object	
__class__	

15.14.3   Class Variables

Name	Description
NUM_STARS	**Value:** 30

Name	Description
WHITE	**Value:** (255, 255, 255)
LEFT	**Value:** 0
RIGHT	**Value:** 1
direction	**Value:** 0
stars	**Value:** []

15.15   Class TextInput

object ─┐
         pygame-asteroids.virtual_keyboard.TextInput

Handles the text input box and manages the cursor

15.15.1   Methods

__init__(*self, background, screen, font, text, x, y*)

x.__init__(...) initializes x; see help(type(x)) for signature

Overrides: object.__init__ extit(inherited documentation)

---

draw(*self*)

Draw the text input box

---

flashcursor(*self*)

Toggle visibility of the cursor

---

addcharatcursor(*self, letter*)

Add a character whereever the cursor is currently located

---

backspace(*self*)

Delete a character before the cursor position

---

deccursor(*self*)

Move the cursor one space left

inccursor(*self*)
Move the cursor one space right (but not beyond the end of the text)

drawcursor(*self*)
Draw the cursor

Inherited from object

__delattr__(), __format__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

15.15.2   Properties

Name	Description
Inherited from object	
__class__	

15.16   Class TouchButtons

15.16.1   Methods

__init__(*self*, *screen*, *button_width*=40)

create_touch_buttons(*self*, *button_width*)

draw(*self*)

detect_actions(*self*, *event*, *actions*)

15.17   Class VirtualKey

object ─┐
        pygame-asteroids.virtual_keyboard.VirtualKey

A single key for the VirtualKeyboard

15.17.1 Methods

__init__(*self, caption, x, y, w*=67, *h*=67)

x.__init__(...) initializes x; see help(type(x)) for signature

Overrides: object.__init__ extit(inherited documentation)

---

draw(*self, screen, background, shifted*=`False`, *forcedraw*=`False`)

Draw one key if it needs redrawing

Inherited from object

__delattr__(), __format__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

15.17.2 Properties

Name	Description
Inherited from object	
__class__	

15.18 Class VirtualKeyboard

object ─┐
        pygame-asteroids.virtual_keyboard.VirtualKeyboard

Implement a basic full screen virtual keyboard for touchscreens

15.18.1 Methods

---

run(*self, screen, font, text*=`''`, *callback*=`None`)

---

not_null(*self, text*)

---

unselectall(*self, force*=`False`)

Force all the keys to be unselected Marks any that change as dirty to redraw

---

---

clickatmouse(*self*)

Check to see if the user is pressing down on a key and draw it selected

---

togglecaps(*self*)

Toggle uppercase / lowercase

---

selectatmouse(*self*)

User has clicked a key, let's use it

---

addkeys(*self*)

Adds the setup for the keys. This would be easy to modify for additional keys

The default start position places the keyboard slightly left of center by design so many people have issues with the right side of their touchscreens that I did this on purpose.

---

paintkeys(*self*)

Draw the keyboard (but only if they're dirty.)

---

clear(*self*)

Put the screen back to before we started

Inherited from object

    __delattr__(), __format__(), __getattribute__(), __hash__(), __init__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

15.18.2 Properties

Name	Description
Inherited from object	
__class__	

15.19   Class error

object ⌐
exceptions.BaseException ⌐
            exceptions.Exception ⌐
                           resource.error

15.19.1   Methods

Inherited from exceptions.Exception

   __init__(), __new__()

Inherited from exceptions.BaseException

   __delattr__(), __getattribute__(), __getitem__(), __getslice__(), __reduce__(), __repr__(),
   __setattr__(), __setstate__(), __str__(), __unicode__()

Inherited from object

   __format__(), __hash__(), __reduce_ex__(), __sizeof__(), __subclasshook__()

15.19.2   Properties

Name	Description
Inherited from exceptions.BaseException	
args, message	
Inherited from object	
__class__	

15.20   Class struct_rusage

object ⌐
        resource.struct_rusage

```
struct_rusage: Result from getrusage.
```

```
This object may be accessed either as a tuple of
    (utime,stime,maxrss,ixrss,idrss,isrss,minflt,majflt,
```

```
    nswap,inblock,oublock,msgsnd,msgrcv,nsignals,nvcsw,nivcsw)
or via the attributes ru_utime, ru_stime, ru_maxrss, and so on.
```

15.20.1 Methods

__add__(*x*, *y*)
x+y

__contains__(*x*, *y*)
y in x

__eq__(*x*, *y*)
x==y

__ge__(*x*, *y*)
x>=y

__getitem__(*x*, *y*)
x[y]

__getslice__(*x*, *i*, *j*)
x[i:j]
Use of negative indices is not supported.

__gt__(*x*, *y*)
x>y

__hash__(*x*)
hash(x)
Overrides: object.__hash__

__le__(*x*, *y*)
x<=y

__len__(*x*)

len(x)

__lt__(*x, y*)

x<y

__mul__(*x, n*)

x*n

__ne__(*x, y*)

x!=y

__new__(*T, S, ...*)

Return Value
     a new object with type S, a subtype of T

Overrides: object.__new__

__reduce__(*...*)

helper for pickle

Overrides: object.__reduce__ extit(inherited documentation)

__repr__(*x*)

repr(x)

Overrides: object.__repr__

__rmul__(*x, n*)

n*x

Inherited from object

__delattr__(), __format__(), __getattribute__(), __init__(), __reduce_ex__(), __setattr__(),
__sizeof__(), __str__(), __subclasshook__()

15.20.2 Properties

Name	Description
ru_idrss	unshared data size
ru_inblock	block input operations

Name	Description
ru_isrss	unshared stack size
ru_ixrss	shared memory size
ru_majflt	page faults requiring I/O
ru_maxrss	max. resident set size
ru_minflt	page faults not requiring I/O
ru_msgrcv	IPC messages received
ru_msgsnd	IPC messages sent
ru_nivcsw	involuntary context switches
ru_nsignals	signals received
ru_nswap	number of swap outs
ru_nvcsw	voluntary context switches
ru_oublock	block output operations
ru_stime	system time used
ru_utime	user time used
Inherited from object	
__class__	

15.20.3 Class Variables

Name	Description
n_fields	**Value:** 16
n_sequence_fields	**Value:** 16
n_unnamed_fields	**Value:** 0

16   Module pygame-asteroids.main

Version: 1.0

Author: Humberto Lino

16.1   Functions

main()
Método principal, que inicializa o jogo

17 Module pygame-asteroids.menu_helper

17.1 Class MenuItem

pygame.font.Font �

pygame-asteroids.menu_helper.MenuItem

Item de Menu

17.1.1 Methods

__init__(*self, text, padding_y*=0, *font_path*=None, *font_size*=Font.SMALL, *font_color*=Color.WHITE, *pos_x_pos_y*=(0,0))

Construtor

Parameters
    text:

    padding_y:

    font_path:

    font_size:

    font_color:

    pos_x_pos_y:

Return Value
    Instância

---

is_mouse_selection(*self, posx_posy*)

Seleção feita pelo mouse?

Parameters
    posx_posy: Posição a verificar

Return Value
    boolean

---

set_position(*self, x, y*)

---

set_font_color(*self, rgb_tuple*)

---

get_font_color(*self*)

17.2 Class Menu

Menu

17.2.1 Methods

__init__(*self, game_controller, items, funcs, padding_y=0, padding_item_y=0, bg_color*=`Color.BLACK`, *select_color*=`Color.GREEN`, *font_path*=`None`, *font_size*=`Font.SMALL`, *font_color*=`Color.WHITE`)

Construtor

Parameters
  game_controller:

  items:

  funcs:

  padding_y:

  padding_item_y:

  bg_color:

  select_color:

  font_path:

  font_size:

  font_color:

Return Value
  Instância

---

back(*self*)

---

set_mouse_visibility(*self*)

---

set_keyboard_selection(*self, key*)

Marks the MenuItem chosen via up and down keys.

---

set_mouse_selection(*self, item, mpos*)

Marks the MenuItem the mouse cursor hovers on.

---

play_hover_sound(*self*)

---

play_enter_sound(*self*)

---

draw(*self*, *draw_callback_func*=None)

Desenhar

Parameters
    draw_callback_func:

---

run(*self*, *fps*, *draw_callback_func*=None)

17.3   Class MainMenu

Menu Principal

17.3.1   Methods

__init__(*self*, *game_controller*, *call_back_function*=None)

Construtor

Parameters
    game_controller:

    call_back_function:

Return Value
    Instância

---

run(*self*)

---

create_main_menu(*self*)

Cria menu principal

---

play(*self*)

Jogar

---

high_score(*self*, *position_to_hightlight*=None)

Placar

---

how_to_play(*self*)

Como Jogar

credits(*self*)

Créditos

18　Module pygame-asteroids.polygon'

18.1　Variables

Name	Description
__package__	**Value: 'pygame-asteroids'**

18.2　Class Color

Representa cor

18.2.1　Methods

---

hex_to_rgb(*value*)

Hexadecimal para RGB

Parameters
　　`value`: Valor hexadecimal

Return Value
　　Cor RGB

---

rgb_to_hex(*rgb*)

Converte cor rgb em hexadecimal

Parameters
　　`rgb`: Cor rgb

Return Value
　　Cor Hexadecimal

---

random_color()

Retorna cor aleatória

Return Value
　　Cor (r,g,b)

---

color_replace(*surface, find_color, replace_color*)

Substitui cor da image

Parameters

    surface: Imagem

    find_color: Cor a procurar

    replace_color: Cor a substituir

Return Value

    Image Modificada

18.2.2   Class Variables

Name	Description
WHITE	Branco **Value:** (255, 255, 255)
RED	Vermelho **Value:** (255, 0, 0)
GREEN	Verde **Value:** (0, 255, 0)
BLACK	Preto **Value:** (0, 0, 0)
YELLOW	Amarelo **Value:** (255, 255, 0)
BLUE	Azul **Value:** (68, 204, 230)
TRANSPARENT	Transparente **Value:** (255, 255, 255, 0)
PINK	Rosa **Value:** (255, 0, 255, 255)
GRAY	Cinza **Value:** (228, 228, 228)

18.3   Class Cube

18.3.1   Methods

__init__(*self*)

---

draw(*self, screen*)

18.4   Class Point3D

18.4.1   Methods

__init__(*self*, *x=0*, *y=0*, *z=0*)

---

project(*self*, *win_width*, *win_height*, *fov*, *viewer_distance*)

Transforms this 3D point to 2D using a perspective projection.

---

rotateX(*self*, *angle*)

Rotates the point around the X axis by the given angle in degrees.

---

rotateY(*self*, *angle*)

Rotates the point around the Y axis by the given angle in degrees.

---

rotateZ(*self*, *angle*)

Rotates the point around the Z axis by the given angle in degrees.

18.5   Class itemgetter

object ¬

        operator.itemgetter

itemgetter(item, ...) –> itemgetter object

Return a callable object that fetches the given item(s) from its operand. After f = itemgetter(2), the call f(r) returns r[2]. After g = itemgetter(2, 5, 3), the call g(r) returns (r[2], r[5], r[3])

18.5.1   Methods

__call__(*x*, *...*)

x(...)

__getattribute__(*...*)

x.__getattribute__('name') <==> x.name

Overrides: object.__getattribute__

__new__(*T, S, ...*)

Return Value
    a new object with type S, a subtype of T

Overrides: object.__new__

Inherited from object

__delattr__(), __format__(), __hash__(), __init__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

18.5.2   Properties

Name	Description
Inherited from object	
__class__	

19 Module pygame-asteroids.resource_manager'

19.1 Variables

Name	Description
__package__	**Value:** 'pygame-asteroids'

19.2 Class Font

19.2.1 Class Variables

Name	Description
LARGE	**Value:** 60
MEDIUM	**Value:** 40
NORMAL	**Value:** 30
SMALL	**Value:** 20
VERY_SMALL	**Value:** 15

19.3 Class ResourceManager

Gerenciador de Recursos

19.3.1 Methods

__init__(*self, dir_root=*None)

Construtor

Parameters
    `dir_root`: Diretório raiz

Return Value
    Instância

---

get_font(*self, size=*30)

Obtém Fonte padrão

Parameters
    `size`: Tamanho

Return Value
    Fonte

---

get_font_by_name(*self, name, size*)

Obtém Fonte pelo nome

Parameters
    `name`: Nome

    `size`: Tamanho

Return Value
    Fonte

---

get_font_default_name(*self*)

Obtém Fonte padrão

Return Value
    Fonte

---

get_font_large(*self, name*)

Obtém Fonte Grande

Parameters
    `name`: Nome

Return Value
    Fonte

---

get_font_normal(*self, name*)

Obtém Fonte Normal

Parameters
    `name`: Nome

Return Value
    Fonte

---

get_font_path(*self, name*)

Obtém caminho da fonte

Parameters
    `name`: Nome

Return Value
    Fonte

get_font_small(*self, name*)

Obtém Fonte pequena

Parameters
    `name`: Nome

Return Value
    Fonte

---

get_font_very_small(*self, name*)

Obtém Fonte muito pequena

Parameters
    `name`: Nome

Return Value
    Fonte

---

get_image(*self, name*)

Obtém imagem

Parameters
    `name`: Nome

Return Value
    Imagem

---

get_scalled_image(*self, name, scale*)

---

init_mixer(*self*)

Inicia mixer

---

list_files(*self, dir*)

---

load(*self, dir_root*)

Carregar

Parameters
    `dir_root`: Diretório raiz

---

load_font(*self*)

Carregar fontes

---

load_image(*self*)

Carregar imagens

---

load_music(*self*)

Carregar efeitos sonoros

---

load_sound(*self*)

Carregar efeitos sonoros

---

play_music(*self*, *name*, *loops*=1)

Reproduz música

Parameters

    `name`: Nome

    `loops`: Quantas vezes repetir

---

play_sound(*self*, *name*, *loops*=1)

Reproduzir efeito sonoro

Parameters

    `name`: Nome

    `loops`: Repetição

19.3.2   Class Variables

Name	Description
fonts	**Value:** {}
images	**Value:** {}
musics	**Value:** {}
sounds	**Value:** {}

20   Module pygame-asteroids.score

20.1   Variables

Name	Description
__package__	**Value:** `None`

20.2   Class DataHandler

20.2.1   Methods

defaultscore(*self*)

21   Module pygame-asteroids.ship

21.1   Variables

Name	Description
RLIMIT_AS	**Value:** 5
RLIMIT_CORE	**Value:** 4
RLIMIT_CPU	**Value:** 0
RLIMIT_DATA	**Value:** 2
RLIMIT_FSIZE	**Value:** 1
RLIMIT_MEMLOCK	**Value:** 6
RLIMIT_NOFILE	**Value:** 8
RLIMIT_NPROC	**Value:** 7
RLIMIT_RSS	**Value:** 5
RLIMIT_STACK	**Value:** 3
RLIM_INFINITY	**Value:** 9223372036854775807
RUSAGE_CHILDREN	**Value:** -1
RUSAGE_SELF	**Value:** 0
__package__	**Value:** 'pygame-asteroids'

21.2   Class Ship

pygame-asteroids.sprite_collision.SpriteCollision

object

pygame.sprite.Sprite

pygame-asteroids.ship.Ship

Nave

21.2.1   Methods

__init__(*self, location, game_controller, *groups*)

Construtor

Parameters
    `location`:

    `game_controller`:

    `groups`:

Return Value
    Instância

Overrides: object.__init__

---

accelerate(*self, angle*)

---

accelerate_speed(*self, angle, x, y*)

Acelerar

Parameters
    `angle`: Ângulo

    `x`:   Posição X

    `y`:   Posição Y

---

shoot(*self*)

Atirar

---

get_center(*self*)

---

explode(*self, time*)

Explidir nave

Parameters
    `time`: Delta Time

---

left(*self, time*)

---

right(*self, time*)

---

up(*self*)

---

rotate(*self, angle*)

---

update(*self, time*)

Atualizar

Parameters
    `time`: Delta Time

Overrides: pygame.sprite.Sprite.update

Inherited from pygame-asteroids.sprite_collision.SpriteCollision(Section 33.12)

collision(), draw2(), get_class_name(), initrect(), rectupdate(), screencollision()

Inherited from pygame.sprite.Sprite

__repr__(), add(), add_internal(), alive(), groups(), kill(), remove(), remove_internal()

Inherited from object

__delattr__(), __format__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

21.2.2 Properties

Name	Description
Inherited from object	
__class__	

21.2.3 Class Variables

Name	Description
SPEED_ACCELERATE	Velociadade de aceleração **Value:** `0.025`
SPEED_ROTATE	Velocidade de rotação **Value:** `0.4`
TIME_LOAD	Tempo para carregar **Value:** `200.0`
TIME_RELOAD_SHOT	Tempo para recarregar **Value:** `100`

22 Module pygame-asteroids.sprite_collision

22.1 Variables

Name	Description
__package__	**Value:** 'pygame-asteroids'

22.2 Class SpriteCollision

Known Subclasses: pygame-asteroids.asteroid.Asteroid, pygame-asteroids.bullet.Bullet, pygame-asteroids.ship.Ship

Classe principal que controla sprite e aplicada a todos os objetos em movimento

22.2.1 Methods

---
initrect(*self, location, game_controller*)

Iniciliza retângulo

Parameters
 location:

 game_controller:

---
get_class_name(*self*)

Retorna nome da classe instanciada

Return Value
 Nome

---
screencollision(*self, time*)

Envolve a tela do jogo para objetos

Parameters
 time: Delta time

---
rectupdate(*self*)

define qual borda colide com o objeto de colisão / blitting

---

draw2(*self, screen*)

blit de imagens para corrigir os lugares quando perto cantos / lados.

Parameters
    `screen`: Tela

---

collision(*self, object*)

Detecta colisão

Parameters
    `object`: Objeto

Return Value
    Booleano

23   Module pygame-asteroids.start_field

23.1   Variables

Name	Description
__package__	**Value:** `'pygame-asteroids'`

23.2   Class StarField

object ⌐

     pygame-asteroids.start_field.StarField

23.2.1   Methods

__init__(*self, screen*)

Create the starfield

Overrides: object.__init__

---

move(*self, start, end, direction*)

Correct for stars hitting the screen's borders

---

fix_stars_on_screen(*self, star*)

---

draw(*self*)

Inherited from object

     __delattr__(), __format__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

23.2.2   Properties

Name	Description
Inherited from object	
__class__	

23.2.3   Class Variables

Name	Description
NUM_STARS	**Value:** 30
WHITE	**Value:** (255, 255, 255)
LEFT	**Value:** 0
RIGHT	**Value:** 1
direction	**Value:** 0
stars	**Value:** []

24   Module pygame-asteroids.state_machine'

Generalização de uma máquina de estado. Utilizada para gerenciar o fluxo do programa

24.1   Variables

Name	Description
__package__	**Value: None**

24.2   Class StateMachine

object ─┐
        └─ **pygame-asteroids.state_machine.StateMachine**

Construtor.

24.2.1   Methods

__init__(*self*)

x.__init__(...) initializes x; see help(type(x)) for signature

Overrides: object.__init__ extit(inherited documentation)

---

draw(*self*, *surface*, *interpolate*)

---

flip_state(*self*)

When a State changes to done necessary startup and cleanup functions are called and the current State is changed.

---

get_event(*self*, *event*)

Pass events down to current State.

---

setup_states(*self*, *state_dict*, *start_state*)

Given a dictionary of states and a state to start in, creates the self.state_dict.

update(*self*, *keys*, *now*)
Checks if a state is done or has called for a game quit. State is flipped if neccessary and State.update is called.

Inherited from object

__delattr__(), __format__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

24.2.2   Properties

Name	Description
Inherited from object	
__class__	

25 Module pygame-asteroids.states.franchise'

Tela como selo de classificação do jogo. Segunda coisa que o jogador visualiza.

25.1 Functions

decode_joystick_arrow_to_keyboard_key(*joystick*)

getpagesize(...)

getrlimit(...)

getrusage(...)

is_event_type_an_input_down(*event*)

setrlimit(...)

vc_get_angle(*center*, *pos*)

Parameters
     pos : posição do mouse
 Return Value
     ângulo em raidanos da posição do mouse em relação ao centro informado. Controle Virtual é dividido em três partes. Para o eixo-y : 1) above the center of the controller 2) below the center of the controller 3) at the same height of the center of the controller If the mouse point is above the center of the controller, than check for one of three conditions: 1) x is to the right of the controller 2) x is to the left of the controller 3) x is at the same point as the centerx of the controller

25.2 Variables

Name	Description
RLIMIT_AS	**Value:** 5
RLIMIT_CORE	**Value:** 4
RLIMIT_CPU	**Value:** 0
RLIMIT_DATA	**Value:** 2
RLIMIT_FSIZE	**Value:** 1

Name	Description
RLIMIT_MEMLOCK	**Value:** 6
RLIMIT_NOFILE	**Value:** 8
RLIMIT_NPROC	**Value:** 7
RLIMIT_RSS	**Value:** 5
RLIMIT_STACK	**Value:** 3
RLIM_INFINITY	**Value:** 9223372036854775807
RUSAGE_CHILDREN	**Value:** -1
RUSAGE_SELF	**Value:** 0
__package__	**Value:** 'pygame-asteroids.states'

25.3 Class Color

Representa cor

25.3.1 Methods

---

color_replace(*surface*, *find_color*, *replace_color*)

Substitui cor da image

Parameters
    surface: Imagem

    find_color: Cor a procurar

    replace_color: Cor a substituir

Return Value
    Image Modificada

---

hex_to_rgb(*value*)

Hexadecimal para RGB

Parameters
    value: Valor hexadecimal

Return Value
    Cor RGB

---

random_color()

Retorna cor aleatória

Return Value
    Cor (r,g,b)

---

rgb_to_hex(*rgb*)

Converte cor rgb em hexadecimal

Parameters
    `rgb`: Cor rgb

Return Value
    Cor Hexadecimal

25.3.2 Class Variables

Name	Description
BLACK	**Value:** (0, 0, 0)
BLUE	**Value:** (68, 204, 230)
GRAY	**Value:** (228, 228, 228)
GREEN	**Value:** (0, 255, 0)
PINK	**Value:** (255, 0, 255, 255)
RED	**Value:** (255, 0, 0)
TRANSPARENT	**Value:** (255, 255, 255, 0)
WHITE	**Value:** (255, 255, 255)
YELLOW	**Value:** (255, 255, 0)

25.4 Class Cube

25.4.1 Methods

__init__(*self*)

---

draw(*self, screen*)

25.5 Class Font

25.5.1 Class Variables

Name	Description
LARGE	**Value:** 60
MEDIUM	**Value:** 40
NORMAL	**Value:** 30
SMALL	**Value:** 20
VERY_SMALL	**Value:** 15

25.6   Class Franchise

object ┐

state_machine._State ┐

pygame-asteroids.states.franchise.Franchise
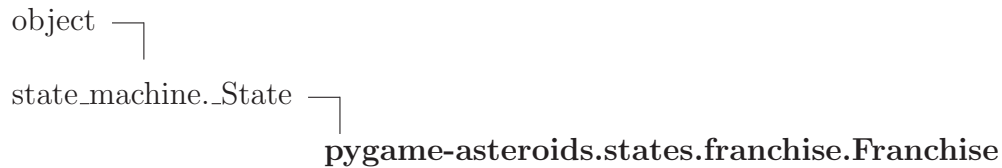
This State is updated while our game shows the splash screen.

25.6.1   Methods

__init__(*self*, *game_controller*)

x.__init__(...) initializes x; see help(type(x)) for signature

Overrides: object.__init__ extit(inherited documentation)

---

draw(*self*, *surface*, *interpolate*)

---

get_event(*self*, *event*)

Get events from Control. Changes to next state on any key press.

Overrides: state_machine._State.get_event

---

update(*self*, *keys*, *now*)

Updates the splash screen.

Overrides: state_machine._State.update

Inherited from state_machine._State

cleanup(), startup()

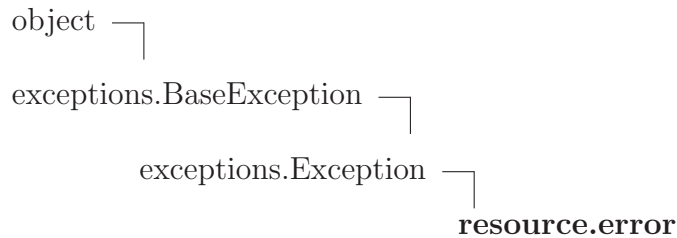### *Inherited from object*

__delattr__(), __format__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

25.6.2   Properties

Name	Description
Inherited from object	
__class__	

25.7 Class Point3D

25.7.1 Methods

__init__(*self, x=0, y=0, z=0*)

---

project(*self, win_width, win_height, fov, viewer_distance*)

Transforms this 3D point to 2D using a perspective projection.

---

rotateX(*self, angle*)

Rotates the point around the X axis by the given angle in degrees.

---

rotateY(*self, angle*)

Rotates the point around the Y axis by the given angle in degrees.

---

rotateZ(*self, angle*)

Rotates the point around the Z axis by the given angle in degrees.

25.8 Class error

object ⌐

exceptions.BaseException ⌐

      exceptions.Exception ⌐

           resource.error
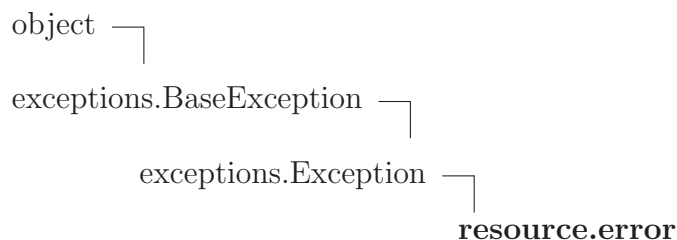
25.8.1 Methods

Inherited from exceptions.Exception

    __init__(), __new__()

Inherited from exceptions.BaseException

    __delattr__(), __getattribute__(), __getitem__(), __getslice__(), __reduce__(), __repr__(), __setattr__(), __setstate__(), __str__(), __unicode__()

Inherited from object

__format__(), __hash__(), __reduce_ex__(), __sizeof__(), __subclasshook__()

25.8.2 Properties

Name	Description
Inherited from exceptions.BaseException	
args, message	
Inherited from object	
__class__	

25.9   Class itemgetter

object ──┐
        operator.itemgetter

itemgetter(item, ...) –> itemgetter object

Return a callable object that fetches the given item(s) from its operand. After f = itemgetter(2), the call f(r) returns r[2]. After g = itemgetter(2, 5, 3), the call g(r) returns (r[2], r[5], r[3])

25.9.1 Methods

__call__(*x, ...*)

x(...)

__getattribute__(...)

x.__getattribute__('name') <==> x.name

Overrides: object.__getattribute__

__new__(*T, S, ...*)

Return Value
    a new object with type S, a subtype of T

Overrides: object.__new__

Inherited from object

__delattr__(), __format__(), __hash__(), __init__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

25.9.2   Properties

Name	Description
Inherited from object	
__class__	

25.10   Class struct_rusage

object ─┐
        │
        └── **resource.struct_rusage**

```
struct_rusage: Result from getrusage.
```

```
This object may be accessed either as a tuple of
    (utime,stime,maxrss,ixrss,idrss,isrss,minflt,majflt,
    nswap,inblock,oublock,msgsnd,msgrcv,nsignals,nvcsw,nivcsw)
or via the attributes ru_utime, ru_stime, ru_maxrss, and so on.
```

25.10.1   Methods

__add__(*x, y*)
x+y

__contains__(*x, y*)
y in x

__eq__(*x, y*)
x==y

__ge__(*x, y*)
x>=y

__getitem__(*x, y*)
x[y]

__getslice__(*x*, *i*, *j*)

x[i:j]

Use of negative indices is not supported.

__gt__(*x*, *y*)

x>y

__hash__(*x*)

hash(x)

Overrides: object.__hash__

__le__(*x*, *y*)

x<=y

__len__(*x*)

len(x)

__lt__(*x*, *y*)

x<y

__mul__(*x*, *n*)

x*n

__ne__(*x*, *y*)

x!=y

__new__(*T*, *S*, ...)

Return Value
     a new object with type S, a subtype of T

Overrides: object.__new__

__reduce__(...)

helper for pickle

Overrides: object.__reduce__ extit(inherited documentation)

__repr__(*x*)

repr(x)

Overrides: object.__repr__

__rmul__(*x*, *n*)

n*x

Inherited from object

__delattr__(), __format__(), __getattribute__(), __init__(), __reduce_ex__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

25.10.2 Properties

Name	Description
ru_idrss	unshared data size
ru_inblock	block input operations
ru_isrss	unshared stack size
ru_ixrss	shared memory size
ru_majflt	page faults requiring I/O
ru_maxrss	max. resident set size
ru_minflt	page faults not requiring I/O
ru_msgrcv	IPC messages received
ru_msgsnd	IPC messages sent
ru_nivcsw	involuntary context switches
ru_nsignals	signals received
ru_nswap	number of swap outs
ru_nvcsw	voluntary context switches
ru_oublock	block output operations
ru_stime	system time used
ru_utime	user time used
Inherited from object	
__class__	

25.10.3 Class Variables

Name	Description
n_fields	**Value:** 16
n_sequence_fields	**Value:** 16
n_unnamed_fields	**Value:** 0

26 Module pygame-asteroids.states.intro'

Tela inicial do jogo. Primeira coisa que o jogador visualiza.

26.1 Functions

decode_joystick_arrow_to_keyboard_key(*joystick*)

is_event_type_an_input_down(*event*)

vc_get_angle(*center*, *pos*)

Parameters
    `pos`: posição do mouse

Return Value
    ângulo em raidanos da posição do mouse em relação ao centro
    informado. Controle Virtual é dividido em três partes. Para o eixo-y
    : 1) above the center of the controller 2) below the center of the
    controller 3) at the same height of the center of the controller If the
    mouse point is above the center of the controller, than check for one
    of three conditions: 1) x is to the right of the controller 2) x is to the
    left of the controller 3) x is at the same point as the centerx of the
    controller

26.2 Variables

Name	Description
__package__	**Value:** `'pygame-asteroids.states'`

26.3 Class Color

Representa cor

26.3.1 Methods

---

color_replace(*surface*, *find_color*, *replace_color*)

Substitui cor da image

Parameters
    surface: Imagem

    find_color: Cor a procurar

    replace_color: Cor a substituir

Return Value
    Image Modificada

---

hex_to_rgb(*value*)

Hexadecimal para RGB

Parameters
    value: Valor hexadecimal

Return Value
    Cor RGB

---

random_color()

Retorna cor aleatória

Return Value
    Cor (r,g,b)

---

rgb_to_hex(*rgb*)

Converte cor rgb em hexadecimal

Parameters
    rgb: Cor rgb

Return Value
    Cor Hexadecimal

26.3.2 Class Variables

Name	Description
BLACK	**Value:** (0, 0, 0)
BLUE	**Value:** (68, 204, 230)
GRAY	**Value:** (228, 228, 228)
GREEN	**Value:** (0, 255, 0)

Name	Description
PINK	**Value:** (255, 0, 255, 255)
RED	**Value:** (255, 0, 0)
TRANSPARENT	**Value:** (255, 255, 255, 0)
WHITE	**Value:** (255, 255, 255)
YELLOW	**Value:** (255, 255, 0)

26.4   Class Font

26.4.1   Class Variables

Name	Description
LARGE	**Value:** 60
MEDIUM	**Value:** 40
NORMAL	**Value:** 30
SMALL	**Value:** 20
VERY_SMALL	**Value:** 15

26.5   Class Introduction

object ⌐

state_machine._State ⌐

pygame-asteroids.states.intro.Introduction
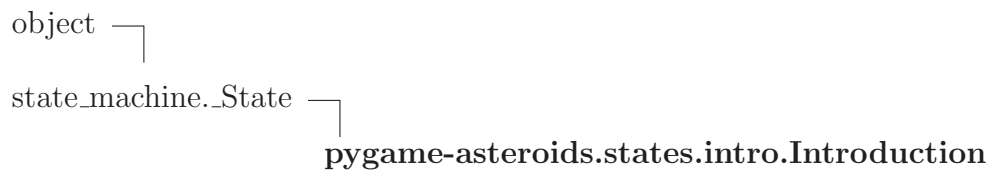
This State is updated while our game shows the splash screen.

26.5.1   Methods

__init__(*self, game_controller*)

x.__init__(...) initializes x; see help(type(x)) for signature

Overrides: object.__init__ extit(inherited documentation)

---

draw(*self, surface, interpolate*)

---

get_event(*self, event*)

Get events from Control. Changes to next state on any key press.

Overrides: state_machine._State.get_event

---

startup(*self, now, persistant*)

Add variables passed in persistant to the proper attributes and set the start time of the State to the current time.

Overrides: state_machine._State.startup extit(inherited documentation)

---

update(*self, keys, now*)

Updates the splash screen.

Overrides: state_machine._State.update

Inherited from state_machine._State

cleanup()

Inherited from object

__delattr__(), __format__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

26.5.2 Properties

Name	Description
Inherited from object	
__class__	

27 Module pygame-asteroids.states.menu

The splash screen of the game. The first thing the user sees.

27.1 Class Menu

object ⌐

pygame-asteroids.state‗machine.‗State ⌐

pygame-asteroids.states.menu.Menu
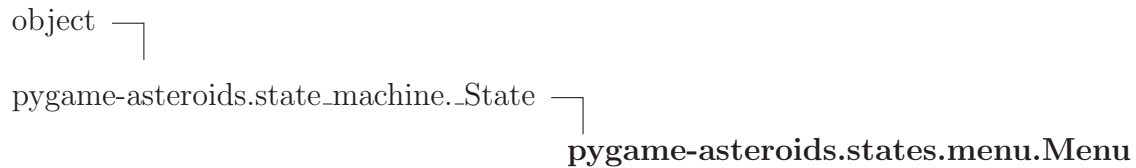
This State is updated while our game shows the splash screen.

27.1.1 Methods

‗‗init‗‗(*self, game‗controller*)

Construtor

Parameters
    `game‗controller`:

Return Value
    Instância

Overrides: object.‗init‗‗

---

update(*self, keys, now*)

Update function for state. Must be overloaded in children.

Overrides: pygame-asteroids.state‗machine.‗State.update extit(inherited documentation)

---

draw(*self, surface, interpolate*)

---

get‗event(*self, event*)

Get events from Control. Changes to next state on any key press.

Overrides: pygame-asteroids.state‗machine.‗State.get‗event

---

startup(*self, now, persistant*)

Ao iniciar State

Parameters

    now:

    persistant:

Overrides: pygame-asteroids.state_machine._State.startup

Inherited from pygame-asteroids.state_machine._State

cleanup()

Inherited from object

__delattr__(), __format__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

27.1.2   Properties

Name	Description
Inherited from object	
__class__	

28 Module pygame-asteroids.states.seal'

Tela como selo de classificação do jogo. Segunda coisa que o jogador visualiza.

28.1 Functions

decode_joystick_arrow_to_keyboard_key(*joystick*)

getpagesize(*...*)

getrlimit(*...*)

getrusage(*...*)

is_event_type_an_input_down(*event*)

setrlimit(*...*)

vc_get_angle(*center*, *pos*)
Parameters   **pos :** posição do mouse
Return Value   ângulo em raidanos da posição do mouse em relação ao centro informado. Controle Virtual é dividido em três partes. Para o eixo-y : 1) above the center of the controller 2) below the center of the controller 3) at the same height of the center of the controller If the mouse point is above the center of the controller, than check for one of three conditions: 1) x is to the right of the controller 2) x is to the left of the controller 3) x is at the same point as the centerx of the controller

28.2 Variables

Name	Description
RLIMIT_AS	**Value:** 5
RLIMIT_CORE	**Value:** 4
RLIMIT_CPU	**Value:** 0
RLIMIT_DATA	**Value:** 2
RLIMIT_FSIZE	**Value:** 1

Name	Description
RLIMIT_MEMLOCK	**Value: 6**
RLIMIT_NOFILE	**Value: 8**
RLIMIT_NPROC	**Value: 7**
RLIMIT_RSS	**Value: 5**
RLIMIT_STACK	**Value: 3**
RLIM_INFINITY	**Value: 9223372036854775807**
RUSAGE_CHILDREN	**Value: -1**
RUSAGE_SELF	**Value: 0**
__package__	**Value: 'pygame-asteroids.states'**

28.3 Class Color

Representa cor

28.3.1 Methods

---

color_replace(*surface*, *find_color*, *replace_color*)

Substitui cor da image

Parameters
     surface: Imagem

     find_color: Cor a procurar

     replace_color: Cor a substituir

Return Value
     Image Modificada

---

hex_to_rgb(*value*)

Hexadecimal para RGB

Parameters
     value: Valor hexadecimal

Return Value
     Cor RGB

---

random_color()

Retorna cor aleatória

Return Value
     Cor (r,g,b)

---

rgb_to_hex(*rgb*)
Converte cor rgb em hexadecimal
Parameters   `rgb`: Cor rgb
Return Value   Cor Hexadecimal

28.3.2   Class Variables

Name	Description
BLACK	**Value:** (0, 0, 0)
BLUE	**Value:** (68, 204, 230)
GRAY	**Value:** (228, 228, 228)
GREEN	**Value:** (0, 255, 0)
PINK	**Value:** (255, 0, 255, 255)
RED	**Value:** (255, 0, 0)
TRANSPARENT	**Value:** (255, 255, 255, 0)
WHITE	**Value:** (255, 255, 255)
YELLOW	**Value:** (255, 255, 0)

28.4   Class Font

28.4.1   Class Variables

Name	Description
LARGE	**Value:** 60
MEDIUM	**Value:** 40
NORMAL	**Value:** 30
SMALL	**Value:** 20
VERY_SMALL	**Value:** 15

28.5   Class Seal

object ┑

state_machine._State ┑

pygame-asteroids.states.seal.Seal

This State is updated while our game shows the splash screen.

28.5.1 Methods

__init__(*self, game_controller*)

x.__init__(...) initializes x; see help(type(x)) for signature

Overrides: object.__init__ extit(inherited documentation)

---

draw(*self, surface, interpolate*)

---

get_event(*self, event*)

Get events from Control. Changes to next state on any key press.

Overrides: state_machine._State.get_event

---

update(*self, keys, now*)

Updates the splash screen.

Overrides: state_machine._State.update

Inherited from state_machine._State

  cleanup(), startup()

Inherited from object

  __delattr__(), __format__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

28.5.2 Properties

Name	Description
Inherited from object	
__class__	

28.6   Class error

object ‐┐

exceptions.BaseException ‐┐

       exceptions.Exception ‐┐

                resource.error

28.6.1   Methods

Inherited from exceptions.Exception

    __init__(), __new__()

Inherited from exceptions.BaseException

    __delattr__(), __getattribute__(), __getitem__(), __getslice__(), __reduce__(), __repr__(),
    __setattr__(), __setstate__(), __str__(), __unicode__()

Inherited from object

    __format__(), __hash__(), __reduce_ex__(), __sizeof__(), __subclasshook__()

28.6.2   Properties

Name	Description
Inherited from exceptions.BaseException	
args, message	
Inherited from object	
__class__	

28.7   Class struct_rusage

object ‐┐

       resource.struct_rusage

```
struct_rusage: Result from getrusage.
```

```
This object may be accessed either as a tuple of
    (utime,stime,maxrss,ixrss,idrss,isrss,minflt,majflt,
```

nswap,inblock,oublock,msgsnd,msgrcv,nsignals,nvcsw,nivcsw)
or via the attributes ru_utime, ru_stime, ru_maxrss, and so on.

28.7.1   Methods

__add__(x, y)

x+y

__contains__(x, y)

y in x

__eq__(x, y)

x==y

__ge__(x, y)

x>=y

__getitem__(x, y)

x[y]

__getslice__(x, i, j)

x[i:j]

Use of negative indices is not supported.

__gt__(x, y)

x>y

__hash__(x)

hash(x)

Overrides: object.__hash__

__le__(x, y)

x<=y

__len__(*x*)

len(x)

__lt__(*x, y*)

x<y

__mul__(*x, n*)

x*n

__ne__(*x, y*)

x!=y

__new__(*T, S, ...*)

Return Value
    a new object with type S, a subtype of T

Overrides: object.__new__

__reduce__(*...*)

helper for pickle

Overrides: object.__reduce__ extit(inherited documentation)

__repr__(*x*)

repr(x)

Overrides: object.__repr__

__rmul__(*x, n*)

n*x

Inherited from object

__delattr__(), __format__(), __getattribute__(), __init__(), __reduce_ex__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

28.7.2   Properties

Name	Description
ru_idrss	unshared data size
ru_inblock	block input operations

Name	Description
ru_isrss	unshared stack size
ru_ixrss	shared memory size
ru_majflt	page faults requiring I/O
ru_maxrss	max. resident set size
ru_minflt	page faults not requiring I/O
ru_msgrcv	IPC messages received
ru_msgsnd	IPC messages sent
ru_nivcsw	involuntary context switches
ru_nsignals	signals received
ru_nswap	number of swap outs
ru_nvcsw	voluntary context switches
ru_oublock	block output operations
ru_stime	system time used
ru_utime	user time used
Inherited from object	
__class__	

28.7.3 Class Variables

Name	Description
n_fields	**Value:** 16
n_sequence_fields	**Value:** 16
n_unnamed_fields	**Value:** 0

29 Package pygame-asteroids.states'

29.1 Modules

- **franchise'**: Tela como selo de classificação do jogo.
  (Section 25, p. 152)
- **intro'**: Tela inicial do jogo.
  (Section 26, p. 161)
- **seal'**: Tela como selo de classificação do jogo.
  (Section 28, p. 167)

29.2 Variables

Name	Description
__package__	**Value:** None

30   Module pygame-asteroids.test_audio'

30.1   Functions

---

decode_joystick_arrow_to_keyboard_key(*joystick*)

---

getpagesize(*...*)

---

getrlimit(*...*)

---

getrusage(*...*)

---

is_event_type_an_input_down(*event*)

---

main()

---

setrlimit(*...*)

---

vc_get_angle(*center*, *pos*)

Parameters
pos: posição do mouse

Return Value
ângulo em raidanos da posição do mouse em relação ao centro informado. Controle Virtual é dividido em três partes. Para o eixo-y : 1) above the center of the controller 2) below the center of the controller 3) at the same height of the center of the controller If the mouse point is above the center of the controller, than check for one of three conditions: 1) x is to the right of the controller 2) x is to the left of the controller 3) x is at the same point as the centerx of the controller

---

30.2   Variables

Name	Description
RLIMIT_AS	Value: 5
RLIMIT_CORE	Value: 4
RLIMIT_CPU	Value: 0
RLIMIT_DATA	Value: 2
RLIMIT_FSIZE	Value: 1
RLIMIT_MEMLOCK	Value: 6

Name	Description
RLIMIT_NOFILE	**Value:** 8
RLIMIT_NPROC	**Value:** 7
RLIMIT_RSS	**Value:** 5
RLIMIT_STACK	**Value:** 3
RLIM_INFINITY	**Value:** 9223372036854775807
RUSAGE_CHILDREN	**Value:** -1
RUSAGE_SELF	**Value:** 0
__package__	**Value:** 'pygame-asteroids'

30.3 Class Asteroid

pygame-asteroids.sprite_collision.SpriteCollision ——

object ——

pygame.sprite.Sprite ——

pygame-asteroids.asteroid.Asteroid
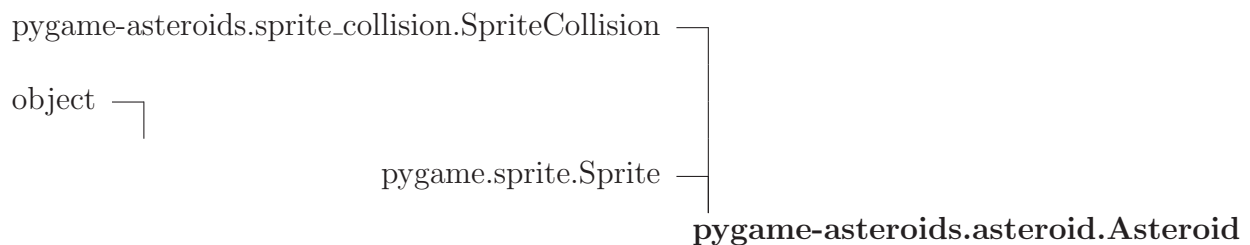
Representa o asteroid

30.3.1 Methods

__init__(*self, location, size, angle, game_controller, from_color*=(255, 0, 255, 255), *to_color*=(255, 0, 0, 255), **groups*)

Construtor

Overrides: object.__init__

---

destroy(*self, angle*)

Destruir asteroids

Parameters
    `angle`: Ângulo

---

random_angle(*self*)

Retorna angulo aleatória

---

rotate(*self*)

---

scale(*self, size*)

---

update(*self, time*)

Atualizar

Parameters
    `time`: Delta Time

Overrides: pygame.sprite.Sprite.update

Inherited from pygame-asteroids.sprite_collision.SpriteCollision(Section 33.12)

collision(), draw2(), get_class_name(), initrect(), rectupdate(), screencollision()

Inherited from pygame.sprite.Sprite

__repr__(), add(), add_internal(), alive(), groups(), kill(), remove(), remove_internal()

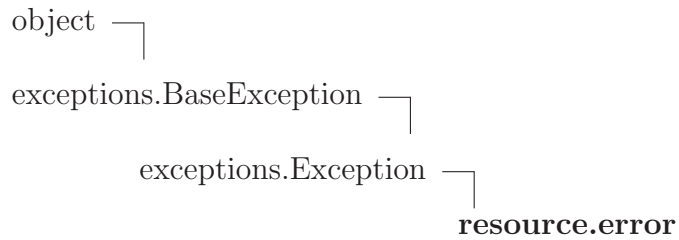### *Inherited from object*

__delattr__(), __format__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

30.3.2   Properties

Name	Description
Inherited from object	
__class__	

30.3.3   Class Variables

Name	Description
SECTIONS	**Value:** 2
SPEED	**Value:** 0.5

30.4   Class Bullet

pygame-asteroids.sprite_collision.SpriteCollision ⎯

object ⎤

                             pygame.sprite.Sprite ⎯

                                       pygame-asteroids.bullet.Bullet
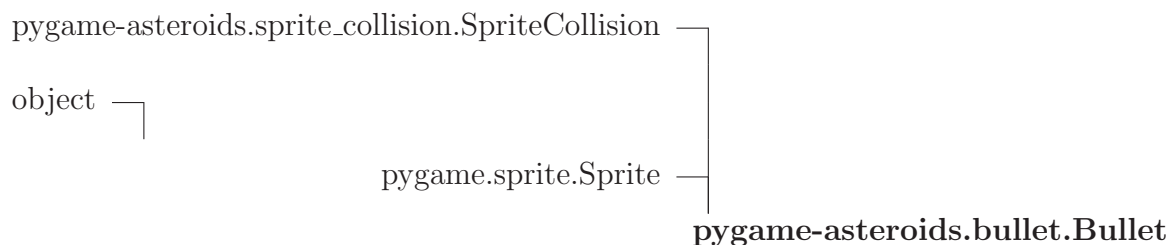
Bala

30.4.1 Methods

__init__(*self, location, angle, game_controller, *groups*)

Construtor

Parameters
    `location:`

    `angle:`

    `game_controller:`

    `groups:`

Return Value
    Instância

Overrides: object.__init__

---

destroy(*self*)

Destruir

---

update(*self, time*)

Atualizar

Parameters
    `time:` Delta time

Overrides: pygame.sprite.Sprite.update

Inherited from pygame-asteroids.sprite_collision.SpriteCollision(Section 33.12)

collision(), draw2(), get_class_name(), initrect(), rectupdate(), screencollision()

Inherited from pygame.sprite.Sprite

__repr__(), add(), add_internal(), alive(), groups(), kill(), remove(), remove_internal()

Inherited from object

__delattr__(), __format__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

30.4.2 Properties

Name	Description
Inherited from object	
__class__	

30.5   Class Color

Representa cor

30.5.1   Methods

---
hex_to_rgb(*value*)

Hexadecimal para RGB

Parameters
   value: Valor hexadecimal

Return Value
   Cor RGB

---
rgb_to_hex(*rgb*)

Converte cor rgb em hexadecimal

Parameters
   rgb: Cor rgb

Return Value
   Cor Hexadecimal

---
random_color()

Retorna cor aleatória

Return Value
   Cor (r,g,b)

---

color_replace(*surface, find_color, replace_color*)

Substitui cor da image

Parameters

  `surface:`   Imagem

  `find_color:`   Cor a procurar

  `replace_color:` Cor a substituir

Return Value

  Image Modificada

30.5.2   Class Variables

Name	Description
WHITE	Branco **Value:** (255, 255, 255)
RED	Vermelho **Value:** (255, 0, 0)
GREEN	Verde **Value:** (0, 255, 0)
BLACK	Preto **Value:** (0, 0, 0)
YELLOW	Amarelo **Value:** (255, 255, 0)
BLUE	Azul **Value:** (68, 204, 230)
TRANSPARENT	Transparente **Value:** (255, 255, 255, 0)
PINK	Rosa **Value:** (255, 0, 255, 255)
GRAY	Cinza **Value:** (228, 228, 228)

30.6   Class DataHandler

30.6.1   Methods

---

defaultscore(*self*)

30.7 Class FileManager

30.7.1 Methods

---
get_pickle_fixed_protocol(*self*)

---
load(*self*, *file_name*, *file_format*={'asteroide': -1, 'controle': 1, 'janela': 2, 'mostrar_mo...*, *sep*=' ')

Carregar arquivo

Parameters

    file_name: Nome do arquivo

    file_format: Formato do arquivo

    sep: Separador

Return Value

    Mapa preenchido

---
marshal(*self*, *file_name*, *object*)

---
unmarshal(*self*, *file_name*)

---
unmarshal_marshal(*self*, *file_name*, *default_object*)

30.7.2 Class Variables

Name	Description
CHARSET	**Value:** 'utf-8'

30.8 Class Font

30.8.1 Class Variables

Name	Description
LARGE	**Value:** 60
MEDIUM	**Value:** 40
NORMAL	**Value:** 30
SMALL	**Value:** 20
VERY_SMALL	**Value:** 15

30.9   Class Game

30.9.1   Methods

__init__(*self, map*)

---
create_empty_actions()

30.10   Class GameController

object ─┐
    pygame-asteroids.game.GameController

Controlador do Jogo

30.10.1   Methods

__init__(*self, game_name, screen, file_manager, asset_manager, config, score_data*)

Construtor

Parameters
    screen:

    file_manager:

    asset_manager:

    config:

    score_data:

Overrides: object.__init__

---
announcelevel(*self, time, screen*)

Mostrar na tela o nível

Parameters
    time: Delta time

    screen: Tela

---
create_asteroids(*self*)

Criar asteroids

---

create_random_asteroids(*self, count*)

Cria asteroids aleatoriamente

Parameters
    self:

Return Value
    asteroids

---

draw_game_over(*self, screen, event, time*)

Desenhar fim de jogo

Parameters
    screen: Tela

    event: Evento

    time: Delta Time

---

draw_hud(*self, screen, time*)

Desenha head-up display

Parameters
    screen: Tela

---

get_display_center(*self*)

---

get_user_score_on_ranking(*self*)

---

initialize_joystick(*self*)

---

main(*self, screen*)

---

quit(*self*)

---

random_position(*self*)

---

save_score(*self, user_name*)

---

screen_rect(*self*)

Retangulo com o tamanho da tela

Return Value
    Retangulo

show_fps(*self*)
Display the current FPS in the window handle if fps_visible is True.

Inherited from object

__delattr__(), __format__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()
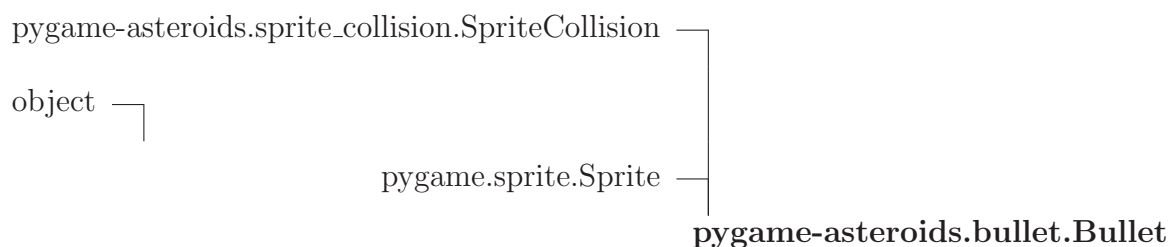
30.10.2 Properties

Name	Description
Inherited from object	
__class__	

30.10.3 Class Variables

Name	Description
newhiscore	**Value:** 0
running	**Value:** 0

30.11 Class ResourceManager

Gerenciador de Recursos

30.11.1 Methods

__init__(*self*, *dir_root*=None)
Construtor
Parameters   **dir_root**: Diretório raiz
Return Value   Instância

get_font(*self, size*=30)

Obtém Fonte padrão

Parameters
    `size`: Tamanho

Return Value
    Fonte

---

get_font_by_name(*self, name, size*)

Obtém Fonte pelo nome

Parameters
    `name`: Nome

    `size`: Tamanho

Return Value
    Fonte

---

get_font_default_name(*self*)

Obtém Fonte padrão

Return Value
    Fonte

---

get_font_large(*self, name*)

Obtém Fonte Grande

Parameters
    `name`: Nome

Return Value
    Fonte

---

get_font_normal(*self, name*)

Obtém Fonte Normal

Parameters
    `name`: Nome

Return Value
    Fonte

get_font_path(*self, name*)

Obtém caminho da fonte

Parameters
    name: Nome

Return Value
    Fonte

---

get_font_small(*self, name*)

Obtém Fonte pequena

Parameters
    name: Nome

Return Value
    Fonte

---

get_font_very_small(*self, name*)

Obtém Fonte muito pequena

Parameters
    name: Nome

Return Value
    Fonte

---

get_image(*self, name*)

Obtém imagem

Parameters
    name: Nome

Return Value
    Imagem

---

get_scalled_image(*self, name, scale*)

---

init_mixer(*self*)

Inicia mixer

---

list_files(*self, dir*)

---

load(*self, dir_root*)

Carregar

Parameters
    `dir_root`: Diretório raiz

---

load_font(*self*)

Carregar fontes

---

load_image(*self*)

Carregar imagens

---

load_music(*self*)

Carregar efeitos sonoros

---

load_sound(*self*)

Carregar efeitos sonoros

---

play_music(*self, name, loops=1*)

Reproduz música

Parameters
    `name`: Nome

    `loops`: Quantas vezes repetir

---

play_sound(*self, name, loops=1*)

Reproduzir efeito sonoro

Parameters
    `name`: Nome

    `loops`: Repetição

30.11.2   Class Variables

Name	Description
fonts	**Value:** {}
images	**Value:** {}
musics	**Value:** {}
sounds	**Value:** {}

30.12   Class Ship

pygame-asteroids.sprite_collision.SpriteCollision ┐

object ┐

pygame.sprite.Sprite ┐

pygame-asteroids.ship.Ship
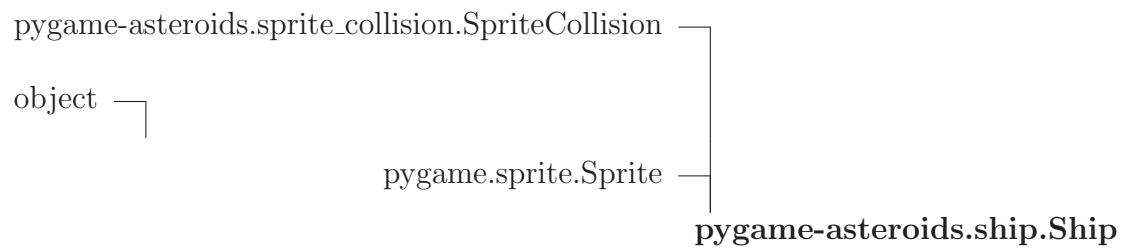
Nave

30.12.1   Methods

__init__(*self, location, game_controller, *groups*)

Construtor

Parameters
    `location`:

    `game_controller`:

    `groups`:

Return Value
    Instância

Overrides: object.__init__

---

accelerate(*self, angle*)

---

accelerate_speed(*self, angle, x, y*)

Acelerar

Parameters
    `angle`: Ângulo

    `x`:   Posição X

    `y`:   Posição Y

---

shoot(*self*)

Atirar

---

get_center(*self*)

---

explode(*self, time*)

Explidir nave

Parameters
    time: Delta Time

---

left(*self, time*)

---

right(*self, time*)

---

up(*self*)

---

rotate(*self, angle*)

---

update(*self, time*)

Atualizar

Parameters
    time: Delta Time

Overrides: pygame.sprite.Sprite.update

Inherited from pygame-asteroids.sprite_collision.SpriteCollision(Section 33.12)

collision(), draw2(), get_class_name(), initrect(), rectupdate(), screencollision()

Inherited from pygame.sprite.Sprite

__repr__(), add(), add_internal(), alive(), groups(), kill(), remove(), remove_internal()

Inherited from object

__delattr__(), __format__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

30.12.2   Properties

Name	Description
Inherited from object	
__class__	

30.12.3   Class Variables

Name	Description
SPEED_ACCELERATE	Velociadade de aceleração **Value:** `0.025`
SPEED_ROTATE	Velocidade de rotação **Value:** `0.4`
TIME_LOAD	Tempo para carregar **Value:** `200.0`
TIME_RELOAD_SHOT	Tempo para recarregar **Value:** `100`

30.13   Class SpriteCollision

Known Subclasses: pygame-asteroids.asteroid.Asteroid, pygame-asteroids.bullet.Bullet, pygame-asteroids.ship.Ship

Classe principal que controla sprite e aplicada a todos os objetos em movimento

30.13.1   Methods

---

initrect(*self, location, game_controller*)

Iniciliza retângulo

Parameters
    `location:`

    `game_controller:`

---

get_class_name(*self*)

Retorna nome da classe instanciada

Return Value
    Nome

---

screencollision(*self, time*)

Envolve a tela do jogo para objetos

Parameters
    `time:` Delta time

---

rectupdate(*self*)

define qual borda colide com o objeto de colisão / blitting

---

draw2(*self, screen*)

---

blit de imagens para corrigir os lugares quando perto cantos / lados.

Parameters
    `screen`: Tela

---

collision(*self, object*)

Detecta colisão

Parameters
    `object`: Objeto

Return Value
    Booleano

30.14 Class SpriteGroup

object ⌐

pygame.sprite.AbstractGroup ⌐

        pygame.sprite.Group ⌐

                            pygame-asteroids.game.SpriteGroup
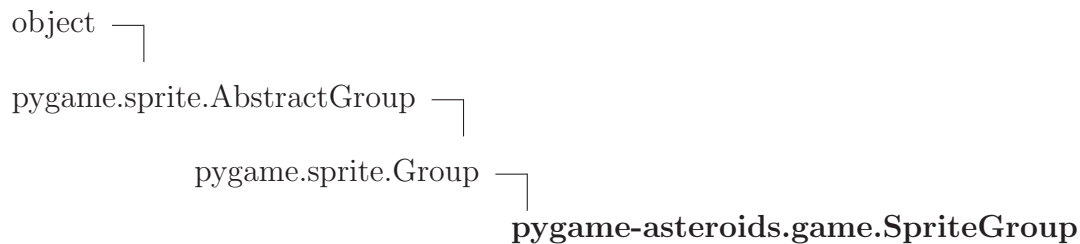
Agrupado de Sprites

30.14.1 Methods

---

draw2(*self, screen*)

Desenhar na tela

Parameters
    `screen`: Tela

Inherited from pygame.sprite.Group

    __init__()

Inherited from pygame.sprite.AbstractGroup

    __contains__(), __iter__(), __len__(), __nonzero__(), __repr__(), add(), add_internal(),
    clear(), copy(), draw(), empty(), has(), has_internal(), remove(), remove_internal(),
    sprites(), update()

Inherited from object

__delattr__(), __format__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(),
__setattr__(), __sizeof__(), __str__(), __subclasshook__()

30.14.2 Properties

Name	Description
Inherited from object	
__class__	

30.15 Class StarField

object ─┐
   pygame-asteroids.start_field.StarField

30.15.1 Methods

__init__(*self, screen*)

Create the starfield

Overrides: object.__init__

move(*self, start, end, direction*)

Correct for stars hitting the screen's borders

fix_stars_on_screen(*self, star*)

draw(*self*)

Inherited from object

__delattr__(), __format__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(),
__repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

30.15.2 Properties

Name	Description
Inherited from object	
__class__	

30.15.3 Class Variables

Name	Description
NUM_STARS	**Value:** 30
WHITE	**Value:** (255, 255, 255)
LEFT	**Value:** 0
RIGHT	**Value:** 1
direction	**Value:** 0
stars	**Value:** []

30.16 Class TextInput

object ─┐
          pygame-asteroids.virtual_keyboard.TextInput

Handles the text input box and manages the cursor

30.16.1 Methods

__init__(*self, background, screen, font, text, x, y*)

x.__init__(...) initializes x; see help(type(x)) for signature

Overrides: object.__init__ extit(inherited documentation)

---

draw(*self*)

Draw the text input box

---

flashcursor(*self*)

Toggle visibility of the cursor

---

addcharatcursor(*self, letter*)

Add a character whereever the cursor is currently located

---

backspace(*self*)

Delete a character before the cursor position

---

deccursor(*self*)

Move the cursor one space left

---

inccursor(*self*)

Move the cursor one space right (but not beyond the end of the text)

---

drawcursor(*self*)

Draw the cursor

Inherited from object

__delattr__(), __format__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

30.16.2   Properties

Name	Description
Inherited from object	
__class__	

30.17   Class TouchButtons

30.17.1   Methods

__init__(*self, screen, button_width=40*)

---

create_touch_buttons(*self, button_width*)

---

draw(*self*)

---

detect_actions(*self, event, actions*)

30.18   Class VirtualKey

object ──┐

       pygame-asteroids.virtual_keyboard.VirtualKey

A single key for the VirtualKeyboard

30.18.1 Methods

__init__(*self, caption, x, y, w=*67, *h=*67)

x.__init__(...) initializes x; see help(type(x)) for signature

Overrides: object.__init__ extit(inherited documentation)

---

draw(*self, screen, background, shifted=*False, *forcedraw=*False)

Draw one key if it needs redrawing

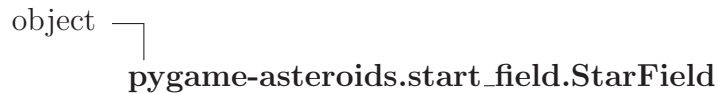### *Inherited from object*

__delattr__(), __format__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

30.18.2 Properties

Name	Description
Inherited from object	
__class__	

30.19 Class VirtualKeyboard

object ─┐

        pygame-asteroids.virtual_keyboard.VirtualKeyboard

Implement a basic full screen virtual keyboard for touchscreens

30.19.1 Methods

---

run(*self, screen, font, text=*'', *callback=*None)

---

not_null(*self, text*)

---

unselectall(*self, force=*False)

Force all the keys to be unselected Marks any that change as dirty to redraw

---

---

clickatmouse(*self*)

Check to see if the user is pressing down on a key and draw it selected

---

togglecaps(*self*)

Toggle uppercase / lowercase

---

selectatmouse(*self*)

User has clicked a key, let's use it

---

addkeys(*self*)

Adds the setup for the keys. This would be easy to modify for additional keys

The default start position places the keyboard slightly left of center by design so many people have issues with the right side of their touchscreens that I did this on purpose.

---

paintkeys(*self*)

Draw the keyboard (but only if they're dirty.)

---

clear(*self*)

Put the screen back to before we started

Inherited from object

__delattr__(), __format__(), __getattribute__(), __hash__(), __init__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

30.19.2 Properties

Name	Description
Inherited from object	
__class__	

30.20   Class error

object ⌐

exceptions.BaseException ⌐

exceptions.Exception ⌐

resource.error

30.20.1   Methods

Inherited from exceptions.Exception

__init__(), __new__()

Inherited from exceptions.BaseException

__delattr__(), __getattribute__(), __getitem__(), __getslice__(), __reduce__(), __repr__(), __setattr__(), __setstate__(), __str__(), __unicode__()

Inherited from object

__format__(), __hash__(), __reduce_ex__(), __sizeof__(), __subclasshook__()

30.20.2   Properties

Name	Description
Inherited from exceptions.BaseException	
args, message	
Inherited from object	
__class__	

30.21   Class struct_rusage

object ⌐

resource.struct_rusage

struct_rusage: Result from getrusage.

This object may be accessed either as a tuple of
    (utime,stime,maxrss,ixrss,idrss,isrss,minflt,majflt,

```
    nswap,inblock,oublock,msgsnd,msgrcv,nsignals,nvcsw,nivcsw)
or via the attributes ru_utime, ru_stime, ru_maxrss, and so on.
```

30.21.1 Methods

__add__(*x, y*)

x+y

__contains__(*x, y*)

y in x

__eq__(*x, y*)

x==y

__ge__(*x, y*)

x>=y

__getitem__(*x, y*)

x[y]

__getslice__(*x, i, j*)

x[i:j]

Use of negative indices is not supported.

__gt__(*x, y*)

x>y

__hash__(*x*)

hash(x)

Overrides: object.__hash__

__le__(*x, y*)

x<=y

__len__(*x*)

len(x)

__lt__(*x, y*)

x<y

__mul__(*x, n*)

x*n

__ne__(*x, y*)

x!=y

__new__(*T, S, ...*)

Return Value
    a new object with type S, a subtype of T

Overrides: object.__new__

__reduce__(*...*)

helper for pickle

Overrides: object.__reduce__ extit(inherited documentation)

__repr__(*x*)

repr(x)

Overrides: object.__repr__

__rmul__(*x, n*)

n*x

Inherited from object

__delattr__(), __format__(), __getattribute__(), __init__(), __reduce_ex__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

30.21.2 Properties

Name	Description
ru_idrss	unshared data size
ru_inblock	block input operations

Name	Description
ru_isrss	unshared stack size
ru_ixrss	shared memory size
ru_majflt	page faults requiring I/O
ru_maxrss	max. resident set size
ru_minflt	page faults not requiring I/O
ru_msgrcv	IPC messages received
ru_msgsnd	IPC messages sent
ru_nivcsw	involuntary context switches
ru_nsignals	signals received
ru_nswap	number of swap outs
ru_nvcsw	voluntary context switches
ru_oublock	block output operations
ru_stime	system time used
ru_utime	user time used
Inherited from object	
__class__	

30.21.3 Class Variables

Name	Description
n_fields	**Value:** 16
n_sequence_fields	**Value:** 16
n_unnamed_fields	**Value:** 0

31 Module pygame-asteroids.test_rotation'

31.1 Functions

---

decode_joystick_arrow_to_keyboard_key(*joystick*)

---

getpagesize(*...*)

---

getrlimit(*...*)

---

getrusage(*...*)

---

is_event_type_an_input_down(*event*)

---

main()

---

setrlimit(*...*)

---

vc_get_angle(*center*, *pos*)

Parameters
    `pos`: posição do mouse

Return Value
    ângulo em raidanos da posição do mouse em relação ao centro informado. Controle Virtual é dividido em três partes. Para o eixo-y : 1) above the center of the controller 2) below the center of the controller 3) at the same height of the center of the controller If the mouse point is above the center of the controller, than check for one of three conditions: 1) x is to the right of the controller 2) x is to the left of the controller 3) x is at the same point as the centerx of the controller

---

31.2 Variables

Name	Description
RLIMIT_AS	Value: 5
RLIMIT_CORE	Value: 4
RLIMIT_CPU	Value: 0
RLIMIT_DATA	Value: 2
RLIMIT_FSIZE	Value: 1
RLIMIT_MEMLOCK	Value: 6

Name	Description
RLIMIT_NOFILE	**Value:** 8
RLIMIT_NPROC	**Value:** 7
RLIMIT_RSS	**Value:** 5
RLIMIT_STACK	**Value:** 3
RLIM_INFINITY	**Value:** 9223372036854775807
RUSAGE_CHILDREN	**Value:** -1
RUSAGE_SELF	**Value:** 0
__package__	**Value:** 'pygame-asteroids'

31.3 Class Asteroid

pygame-asteroids.sprite_collision.SpriteCollision ⌐

object ⌐

pygame.sprite.Sprite ⌐

pygame-asteroids.asteroid.Asteroid
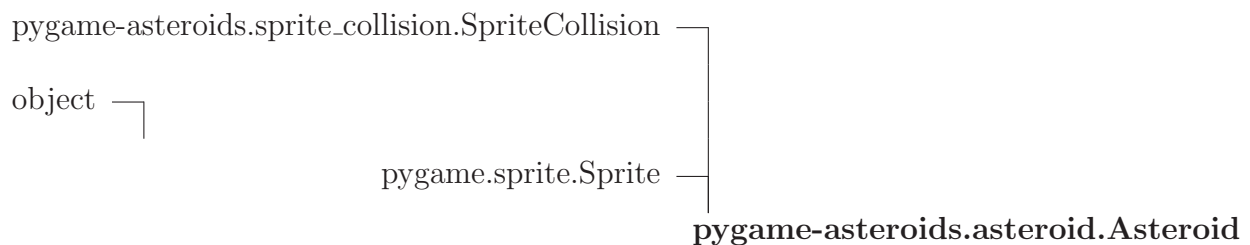
Representa o asteroid

31.3.1 Methods

__init__(*self, location, size, angle, game_controller, from_color*=(255, 0, 255, 255), *to_color*=(255, 0, 0, 255), **groups*)

Construtor

Overrides: object.__init__

---

destroy(*self, angle*)

Destruir asteroids

Parameters
    `angle`: Ângulo

---

random_angle(*self*)

Retorna angulo aleatória

---

rotate(*self*)

---

scale(*self, size*)

---

update(*self, time*)

Atualizar

Parameters
    `time`: Delta Time

Overrides: pygame.sprite.Sprite.update

Inherited from pygame-asteroids.sprite_collision.SpriteCollision(Section 33.12)

collision(), draw2(), get_class_name(), initrect(), rectupdate(), screencollision()

Inherited from pygame.sprite.Sprite

__repr__(), add(), add_internal(), alive(), groups(), kill(), remove(), remove_internal()

Inherited from object

__delattr__(), __format__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

31.3.2   Properties

Name	Description
Inherited from object	
__class__	

31.3.3   Class Variables

Name	Description
SECTIONS	**Value:** 2
SPEED	**Value:** `0.5`

31.4   Class Bullet

pygame-asteroids.sprite_collision.SpriteCollision ─┐

object ─┐

                      pygame.sprite.Sprite ─┐

                                  pygame-asteroids.bullet.Bullet

Bala

31.4.1 Methods

__init__(*self, location, angle, game_controller, *groups*)

Construtor

Parameters
    location:

    angle:

    game_controller:

    groups:

Return Value
    Instância

Overrides: object.__init__

---

destroy(*self*)

Destruir

---

update(*self, time*)

Atualizar

Parameters
    time: Delta time

Overrides: pygame.sprite.Sprite.update

Inherited from pygame-asteroids.sprite_collision.SpriteCollision(Section 33.12)

collision(), draw2(), get_class_name(), initrect(), rectupdate(), screencollision()

Inherited from pygame.sprite.Sprite

__repr__(), add(), add_internal(), alive(), groups(), kill(), remove(), remove_internal()

Inherited from object

__delattr__(), __format__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

31.4.2 Properties

Name	Description
Inherited from object	
__class__	

31.5   Class Color

Representa cor

31.5.1   Methods

---

hex_to_rgb(*value*)

Hexadecimal para RGB

Parameters
    `value`: Valor hexadecimal

Return Value
    Cor RGB

---

rgb_to_hex(*rgb*)

Converte cor rgb em hexadecimal

Parameters
    `rgb`: Cor rgb

Return Value
    Cor Hexadecimal

---

random_color()

Retorna cor aleatória

Return Value
    Cor (r,g,b)

---

color_replace(*surface*, *find_color*, *replace_color*)

Substitui cor da image

Parameters
    `surface:`   Imagem

    `find_color:`   Cor a procurar

    `replace_color:` Cor a substituir

Return Value
    Image Modificada

31.5.2   Class Variables

Name	Description
WHITE	Branco **Value:** (255, 255, 255)
RED	Vermelho **Value:** (255, 0, 0)
GREEN	Verde **Value:** (0, 255, 0)
BLACK	Preto **Value:** (0, 0, 0)
YELLOW	Amarelo **Value:** (255, 255, 0)
BLUE	Azul **Value:** (68, 204, 230)
TRANSPARENT	Transparente **Value:** (255, 255, 255, 0)
PINK	Rosa **Value:** (255, 0, 255, 255)
GRAY	Cinza **Value:** (228, 228, 228)

31.6   Class DataHandler

31.6.1   Methods

defaultscore(*self*)

31.7   Class FileManager

31.7.1   Methods

---
get_pickle_fixed_protocol(*self*)

---
load(*self, file_name, file_format*={'asteroide': -1, 'controle': 1, 'janela': 2, 'mostrar_mo...*, sep*=' '})

Carregar arquivo

Parameters

     file_name: Nome do arquivo

     file_format: Formato do arquivo

     sep: Separador

Return Value

     Mapa preenchido

---
marshal(*self, file_name, object*)

---
unmarshal(*self, file_name*)

---
unmarshal_marshal(*self, file_name, default_object*)

31.7.2   Class Variables

Name	Description
CHARSET	**Value:** 'utf-8'

31.8   Class Font

31.8.1   Class Variables

Name	Description
LARGE	**Value:** 60
MEDIUM	**Value:** 40
NORMAL	**Value:** 30
SMALL	**Value:** 20
VERY_SMALL	**Value:** 15

31.9   Class Game

31.9.1   Methods

__init__(*self, map*)

---
create_empty_actions()

31.10   Class GameController

object ─┐

         pygame-asteroids.game.GameController

Controlador do Jogo

31.10.1   Methods

__init__(*self, game_name, screen, file_manager, asset_manager, config, score_data*)

Construtor

Parameters
     `screen`:

     `file_manager`:

     `asset_manager`:

     `config`:

     `score_data`:

Overrides: object.__init__

---
announcelevel(*self, time, screen*)

Mostrar na tela o nível

Parameters
     `time`:   Delta time

     `screen`: Tela

---
create_asteroids(*self*)

Criar asteroids

---

create_random_asteroids(*self, count*)

Cria asteroids aleatoriamente

Parameters
    `self`:

Return Value
    asteroids

---

draw_game_over(*self, screen, event, time*)

Desenhar fim de jogo

Parameters
    `screen`: Tela

    `event`:   Evento

    `time`:   Delta Time

---

draw_hud(*self, screen, time*)

Desenha head-up display

Parameters
    `screen`: Tela

---

get_display_center(*self*)

---

get_user_score_on_ranking(*self*)

---

initialize_joystick(*self*)

---

main(*self, screen*)

---

quit(*self*)

---

random_position(*self*)

---

save_score(*self, user_name*)

---

screen_rect(*self*)

Retangulo com o tamanho da tela

Return Value
    Retangulo

---

show_fps(*self*)

Display the current FPS in the window handle if fps_visible is True.

Inherited from object

__delattr__(), __format__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

31.10.2 Properties

Name	Description
Inherited from object	
__class__	

31.10.3 Class Variables

Name	Description
newhiscore	**Value:** 0
running	**Value:** 0

31.11 Class ResourceManager

Gerenciador de Recursos

31.11.1 Methods

__**init**__(*self, dir_root=*`None`)

Construtor

Parameters
    `dir_root`: Diretório raiz

Return Value
    Instância

---

get_font(*self, size*=30)

Obtém Fonte padrão

Parameters
    `size`: Tamanho

Return Value
    Fonte

---

get_font_by_name(*self, name, size*)

Obtém Fonte pelo nome

Parameters
    `name`: Nome

    `size`: Tamanho

Return Value
    Fonte

---

get_font_default_name(*self*)

Obtém Fonte padrão

Return Value
    Fonte

---

get_font_large(*self, name*)

Obtém Fonte Grande

Parameters
    `name`: Nome

Return Value
    Fonte

---

get_font_normal(*self, name*)

Obtém Fonte Normal

Parameters
    `name`: Nome

Return Value
    Fonte

get_font_path(*self, name*)

Obtém caminho da fonte

Parameters
    `name`: Nome

Return Value
    Fonte

---

get_font_small(*self, name*)

Obtém Fonte pequena

Parameters
    `name`: Nome

Return Value
    Fonte

---

get_font_very_small(*self, name*)

Obtém Fonte muito pequena

Parameters
    `name`: Nome

Return Value
    Fonte

---

get_image(*self, name*)

Obtém imagem

Parameters
    `name`: Nome

Return Value
    Imagem

---

get_scalled_image(*self, name, scale*)

---

init_mixer(*self*)

Inicia mixer

---

list_files(*self, dir*)

---

load(*self, dir_root*)

Carregar

Parameters
    `dir_root`: Diretório raiz

---

load_font(*self*)

Carregar fontes

---

load_image(*self*)

Carregar imagens

---

load_music(*self*)

Carregar efeitos sonoros

---

load_sound(*self*)

Carregar efeitos sonoros

---

play_music(*self, name, loops=1*)

Reproduz música

Parameters
    `name`: Nome

    `loops`: Quantas vezes repetir

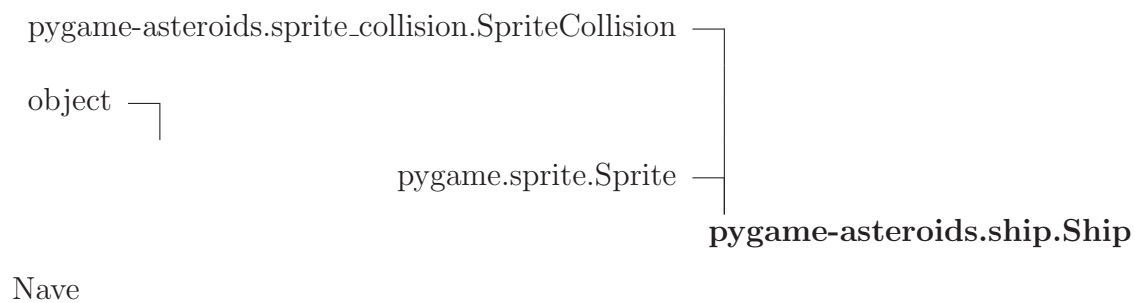**play_sound**(*self, name, loops=1*)

Reproduzir efeito sonoro

Parameters
    `name`: Nome

    `loops`: Repetição

31.11.2 Class Variables

Name	Description
fonts	**Value:** {}
images	**Value:** {}
musics	**Value:** {}
sounds	**Value:** {}

31.12 Class Ship

pygame-asteroids.sprite_collision.SpriteCollision ──┐

object ──┐

                               pygame.sprite.Sprite ──┤

                                        pygame-asteroids.ship.Ship
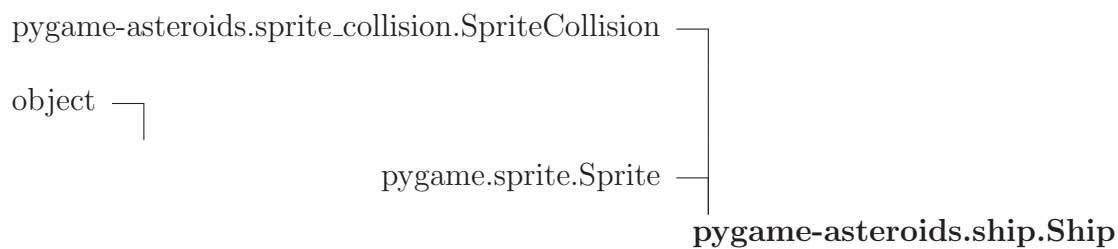
Nave

31.12.1 Methods

__init__(*self, location, game_controller, *groups*)

Construtor

Parameters
    `location`:

    `game_controller`:

    `groups`:

Return Value
    Instância

Overrides: object.__init__

---

accelerate(*self, angle*)

---

accelerate_speed(*self, angle, x, y*)

Acelerar

Parameters
    `angle`: Ângulo

    `x`:   Posição X

    `y`:   Posição Y

---

shoot(*self*)

Atirar

---

get_center(*self*)

---

explode(*self, time*)

Explidir nave

Parameters
    `time`: Delta Time

---

left(*self, time*)

---

right(*self, time*)

---

up(*self*)

---

rotate(*self, angle*)

---

update(*self, time*)

Atualizar

Parameters
    `time`: Delta Time

Overrides: pygame.sprite.Sprite.update

Inherited from pygame-asteroids.sprite_collision.SpriteCollision(Section 33.12)

collision(), draw2(), get_class_name(), initrect(), rectupdate(), screencollision()

Inherited from pygame.sprite.Sprite

__repr__(), add(), add_internal(), alive(), groups(), kill(), remove(), remove_internal()

Inherited from object

__delattr__(), __format__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

31.12.2 Properties

Name	Description
Inherited from object	
__class__	

31.12.3 Class Variables

Name	Description
SPEED_ACCELERATE	Velociadade de aceleração **Value:** `0.025`
SPEED_ROTATE	Velociadade de rotação **Value:** `0.4`
TIME_LOAD	Tempo para carregar **Value:** `200.0`
TIME_RELOAD_SHOT	Tempo para recarregar **Value:** `100`

31.13   Class SpriteCollision

Known Subclasses: pygame-asteroids.asteroid.Asteroid, pygame-asteroids.bullet.Bullet, pygame-asteroids.ship.Ship

Classe principal que controla sprite e aplicada a todos os objetos em movimento

31.13.1   Methods

---

initrect(*self, location, game_controller*)

Iniciliza retângulo

Parameters
    `location:`

    `game_controller:`

---

get_class_name(*self*)

Retorna nome da classe instanciada

Return Value
    Nome

---

screencollision(*self, time*)

Envolve a tela do jogo para objetos

Parameters
    `time:` Delta time

---

rectupdate(*self*)

define qual borda colide com o objeto de colisão / blitting

---

draw2(*self, screen*)

---

blit de imagens para corrigir os lugares quando perto cantos / lados.

Parameters
    `screen`: Tela

---

collision(*self, object*)

Detecta colisão

Parameters
    `object`: Objeto

Return Value
    Booleano

31.14 Class SpriteGroup

object ⌐

pygame.sprite.AbstractGroup ⌐

            pygame.sprite.Group ⌐

                   pygame-asteroids.game.SpriteGroup
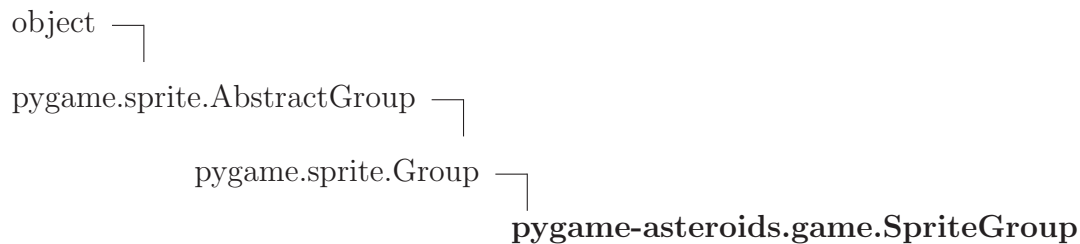
Agrupado de Sprites

31.14.1 Methods

---

draw2(*self, screen*)

Desenhar na tela

Parameters
    `screen`: Tela

Inherited from pygame.sprite.Group

    __init__()

Inherited from pygame.sprite.AbstractGroup

    __contains__(), __iter__(), __len__(), __nonzero__(), __repr__(), add(), add_internal(),
    clear(), copy(), draw(), empty(), has(), has_internal(), remove(), remove_internal(),
    sprites(), update()

Inherited from object

__delattr__(), __format__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

31.14.2   Properties

Name	Description
Inherited from object	
__class__	

31.15   Class StarField

object ─┐

   pygame-asteroids.start_field.StarField

31.15.1   Methods

__init__(*self, screen*)

Create the starfield

Overrides: object.__init__

---
move(*self, start, end, direction*)

Correct for stars hitting the screen's borders

---
fix_stars_on_screen(*self, star*)

---
draw(*self*)

Inherited from object

__delattr__(), __format__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

31.15.2   Properties

Name	Description
Inherited from object	
__class__	

31.15.3   Class Variables

Name	Description
NUM_STARS	**Value:** 30
WHITE	**Value:** (255, 255, 255)
LEFT	**Value:** 0
RIGHT	**Value:** 1
direction	**Value:** 0
stars	**Value:** []

31.16   Class TextInput

object ─┐
          └─
        pygame-asteroids.virtual_keyboard.TextInput

Handles the text input box and manages the cursor

31.16.1   Methods

__**init**__(*self, background, screen, font, text, x, y*)

x.__init__(...) initializes x; see help(type(x)) for signature

Overrides: object.__init__ extit(inherited documentation)

---

draw(*self*)

Draw the text input box

---

flashcursor(*self*)

Toggle visibility of the cursor

---

addcharatcursor(*self, letter*)

Add a character whereever the cursor is currently located

---

backspace(*self*)

Delete a character before the cursor position

---

deccursor(*self*)
Move the cursor one space left

inccursor(*self*)
Move the cursor one space right (but not beyond the end of the text)

drawcursor(*self*)
Draw the cursor

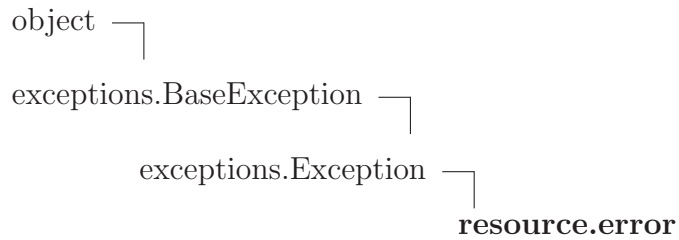### *Inherited from object*

__delattr__(), __format__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

31.16.2   Properties

Name	Description
Inherited from object	
__class__	

31.17   Class TouchButtons

31.17.1   Methods

__init__(*self, screen, button_width=40*)

create_touch_buttons(*self, button_width*)

draw(*self*)

detect_actions(*self, event, actions*)

31.18   Class VirtualKey

object ─┐

   pygame-asteroids.virtual_keyboard.VirtualKey

A single key for the VirtualKeyboard

31.18.1   Methods

__init__(*self*, *caption*, *x*, *y*, *w*=`67`, *h*=`67`)

x.__init__(...) initializes x; see help(type(x)) for signature

Overrides: object.__init__ extit(inherited documentation)

---

draw(*self*, *screen*, *background*, *shifted*=`False`, *forcedraw*=`False`)

Draw one key if it needs redrawing

Inherited from object

__delattr__(), __format__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

31.18.2   Properties

Name	Description
Inherited from object	
__class__	

31.19   Class VirtualKeyboard

object ─┐

   pygame-asteroids.virtual_keyboard.VirtualKeyboard

Implement a basic full screen virtual keyboard for touchscreens

31.19.1   Methods

---

run(*self*, *screen*, *font*, *text*=`''`, *callback*=`None`)

---

not_null(*self*, *text*)

---

unselectall(*self*, *force*=`False`)

Force all the keys to be unselected Marks any that change as dirty to redraw

---

233

clickatmouse(*self*)

Check to see if the user is pressing down on a key and draw it selected

---

togglecaps(*self*)

Toggle uppercase / lowercase

---

selectatmouse(*self*)

User has clicked a key, let's use it

---

addkeys(*self*)

Adds the setup for the keys. This would be easy to modify for additional keys

The default start position places the keyboard slightly left of center by design so many people have issues with the right side of their touchscreens that I did this on purpose.

---

paintkeys(*self*)

Draw the keyboard (but only if they're dirty.)

---

clear(*self*)

Put the screen back to before we started

Inherited from object

__delattr__(), __format__(), __getattribute__(), __hash__(), __init__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

31.19.2 Properties

Name	Description
Inherited from object	
__class__	

31.20   Class error

object ⌐
exceptions.BaseException ⌐
exceptions.Exception ⌐
resource.error
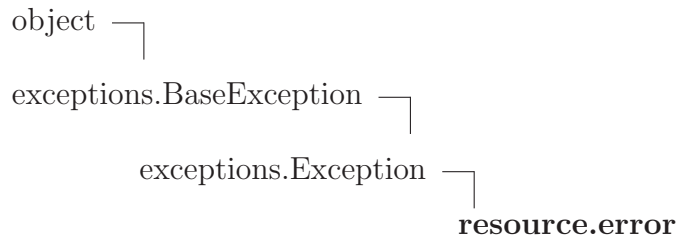
31.20.1   Methods

Inherited from exceptions.Exception

__init__(), __new__()

Inherited from exceptions.BaseException

__delattr__(), __getattribute__(), __getitem__(), __getslice__(), __reduce__(), __repr__(), __setattr__(), __setstate__(), __str__(), __unicode__()

Inherited from object

__format__(), __hash__(), __reduce_ex__(), __sizeof__(), __subclasshook__()

31.20.2   Properties

Name	Description
Inherited from exceptions.BaseException	
args, message	
Inherited from object	
__class__	

31.21   Class struct_rusage

object ⌐
resource.struct_rusage

```
struct_rusage: Result from getrusage.
```

```
This object may be accessed either as a tuple of
    (utime,stime,maxrss,ixrss,idrss,isrss,minflt,majflt,
```

```
    nswap,inblock,oublock,msgsnd,msgrcv,nsignals,nvcsw,nivcsw)
or via the attributes ru_utime, ru_stime, ru_maxrss, and so on.
```

31.21.1 Methods

__add__(*x, y*)

x+y

__contains__(*x, y*)

y in x

__eq__(*x, y*)

x==y

__ge__(*x, y*)

x>=y

__getitem__(*x, y*)

x[y]

__getslice__(*x, i, j*)

x[i:j]

Use of negative indices is not supported.

__gt__(*x, y*)

x>y

__hash__(*x*)

hash(x)

Overrides: object.__hash__

__le__(*x, y*)

x<=y

__len__(*x*)

len(x)

__lt__(*x, y*)

x<y

__mul__(*x, n*)

x*n

__ne__(*x, y*)

x!=y

__new__(*T, S, ...*)

Return Value
    a new object with type S, a subtype of T

Overrides: object.__new__

__reduce__(*...*)

helper for pickle

Overrides: object.__reduce__ extit(inherited documentation)

__repr__(*x*)

repr(x)

Overrides: object.__repr__

__rmul__(*x, n*)

n*x

Inherited from object

__delattr__(), __format__(), __getattribute__(), __init__(), __reduce_ex__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

31.21.2 Properties

Name	Description
ru_idrss	unshared data size
ru_inblock	block input operations

Class struct_rusage

Name	Description
ru_isrss	unshared stack size
ru_ixrss	shared memory size
ru_majflt	page faults requiring I/O
ru_maxrss	max. resident set size
ru_minflt	page faults not requiring I/O
ru_msgrcv	IPC messages received
ru_msgsnd	IPC messages sent
ru_nivcsw	involuntary context switches
ru_nsignals	signals received
ru_nswap	number of swap outs
ru_nvcsw	voluntary context switches
ru_oublock	block output operations
ru_stime	system time used
ru_utime	user time used
Inherited from object	
__class__	

31.21.3 Class Variables

Name	Description
n_fields	**Value:** 16
n_sequence_fields	**Value:** 16
n_unnamed_fields	**Value:** 0

32 Module pygame-asteroids.test_startfield'

32.1 Functions

main()

32.2 Variables

Name	Description
__package__	**Value:** 'pygame-asteroids'

32.3 Class StarField

object ⌐
      pygame-asteroids.start_field.StarField

32.3.1 Methods

__init__(*self, screen*)

Create the starfield

Overrides: object.__init__

---

move(*self, start, end, direction*)

Correct for stars hitting the screen's borders

---

fix_stars_on_screen(*self, star*)

---

draw(*self*)

Inherited from object

    __delattr__(), __format__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

32.3.2 Properties

Name	Description
Inherited from object	
__class__	

32.3.3   Class Variables

Name	Description
NUM_STARS	**Value:** 30
WHITE	**Value:** (255, 255, 255)
LEFT	**Value:** 0
RIGHT	**Value:** 1
direction	**Value:** 0
stars	**Value:** []

33 Module pygame-asteroids.test_touch_buttons'

33.1 Functions

---

decode_joystick_arrow_to_keyboard_key(*joystick*)

---

getpagesize(*...*)

---

getrlimit(*...*)

---

getrusage(*...*)

---

is_event_type_an_input_down(*event*)

---

main()

---

setrlimit(*...*)

---

vc_get_angle(*center*, *pos*)

Parameters
    `pos`: posição do mouse

Return Value
    ângulo em raidanos da posição do mouse em relação ao centro
    informado. Controle Virtual é dividido em três partes. Para o eixo-y
    : 1) above the center of the controller 2) below the center of the
    controller 3) at the same height of the center of the controller If the
    mouse point is above the center of the controller, than check for one
    of three conditions: 1) x is to the right of the controller 2) x is to the
    left of the controller 3) x is at the same point as the centerx of the
    controller

33.2 Variables

Name	Description
RLIMIT_AS	Value: 5
RLIMIT_CORE	Value: 4
RLIMIT_CPU	Value: 0
RLIMIT_DATA	Value: 2
RLIMIT_FSIZE	Value: 1
RLIMIT_MEMLOCK	Value: 6

241

Name	Description
RLIMIT_NOFILE	**Value:** 8
RLIMIT_NPROC	**Value:** 7
RLIMIT_RSS	**Value:** 5
RLIMIT_STACK	**Value:** 3
RLIM_INFINITY	**Value:** 9223372036854775807
RUSAGE_CHILDREN	**Value:** -1
RUSAGE_SELF	**Value:** 0
__package__	**Value:** 'pygame-asteroids'

33.3   Class Asteroid

pygame-asteroids.sprite_collision.SpriteCollision ──

object ─┐

                                        pygame.sprite.Sprite ──

pygame-asteroids.asteroid.Asteroid

Representa o asteroid

33.3.1   Methods

__init__(*self, location, size, angle, game_controller, from_color*=(255, 0, 255, 255), *to_color*=(255, 0, 0, 255), **groups*)

Construtor

Overrides: object.__init__

---

destroy(*self, angle*)

Destruir asteroids

Parameters
    `angle`: Ângulo

---

random_angle(*self*)

Retorna angulo aleatória

---

rotate(*self*)

242

---

scale(*self*, *size*)

---

update(*self*, *time*)

Atualizar

Parameters
    time: Delta Time

Overrides: pygame.sprite.Sprite.update

Inherited from pygame-asteroids.sprite_collision.SpriteCollision(Section 33.12)

collision(), draw2(), get_class_name(), initrect(), rectupdate(), screencollision()

Inherited from pygame.sprite.Sprite

__repr__(), add(), add_internal(), alive(), groups(), kill(), remove(), remove_internal()

Inherited from object

__delattr__(), __format__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

33.3.2 Properties

Name	Description
Inherited from object	
__class__	

33.3.3 Class Variables

Name	Description
SECTIONS	**Value:** 2
SPEED	**Value:** 0.5

33.4 Class Bullet

pygame-asteroids.sprite_collision.SpriteCollision ⎯

object ⎯

                              pygame.sprite.Sprite ⎯

                                        pygame-asteroids.bullet.Bullet
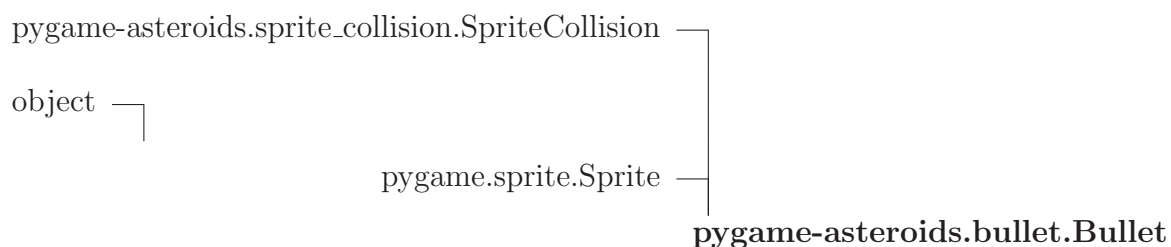
Bala

33.4.1 Methods

__init__(*self, location, angle, game_controller, *groups*)

Construtor

Parameters
    `location:`

    `angle:`

    `game_controller:`

    `groups:`

Return Value
    Instância

Overrides: object.__init__

---

destroy(*self*)

Destruir

---

update(*self, time*)

Atualizar

Parameters
    `time:` Delta time

Overrides: pygame.sprite.Sprite.update

Inherited from pygame-asteroids.sprite_collision.SpriteCollision(Section 33.12)

collision(), draw2(), get_class_name(), initrect(), rectupdate(), screencollision()

Inherited from pygame.sprite.Sprite

__repr__(), add(), add_internal(), alive(), groups(), kill(), remove(), remove_internal()

Inherited from object

__delattr__(), __format__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

33.4.2 Properties

Name	Description
Inherited from object	
__class__	

33.5   Class Color

Representa cor

33.5.1   Methods

---

hex_to_rgb(*value*)

Hexadecimal para RGB

Parameters
    value: Valor hexadecimal

Return Value
    Cor RGB

---

rgb_to_hex(*rgb*)

Converte cor rgb em hexadecimal

Parameters
    rgb: Cor rgb

Return Value
    Cor Hexadecimal

---

random_color()

Retorna cor aleatória

Return Value
    Cor (r,g,b)

---

color_replace(*surface, find_color, replace_color*)

Substitui cor da image

Parameters

    surface: Imagem

    find_color: Cor a procurar

    replace_color: Cor a substituir

Return Value

    Image Modificada

33.5.2   Class Variables

Name	Description
WHITE	Branco **Value:** (255, 255, 255)
RED	Vermelho **Value:** (255, 0, 0)
GREEN	Verde **Value:** (0, 255, 0)
BLACK	Preto **Value:** (0, 0, 0)
YELLOW	Amarelo **Value:** (255, 255, 0)
BLUE	Azul **Value:** (68, 204, 230)
TRANSPARENT	Transparente **Value:** (255, 255, 255, 0)
PINK	Rosa **Value:** (255, 0, 255, 255)
GRAY	Cinza **Value:** (228, 228, 228)

33.6   Class DataHandler

33.6.1   Methods

---

defaultscore(*self*)

33.7   Class FileManager

33.7.1   Methods

---

get_pickle_fixed_protocol(*self*)

---

load(*self*, *file_name*, *file_format*={'asteroide': -1, 'controle': 1, 'janela': 2, 'mostrar_mo...*, *sep*=' ')

Carregar arquivo

Parameters
    file_name: Nome do arquivo

    file_format: Formato do arquivo

    sep: Separador

Return Value
    Mapa preenchido

---

marshal(*self*, *file_name*, *object*)

---

unmarshal(*self*, *file_name*)

---

unmarshal_marshal(*self*, *file_name*, *default_object*)

33.7.2   Class Variables

Name	Description
CHARSET	**Value:** 'utf-8'

33.8   Class Font

33.8.1   Class Variables

Name	Description
LARGE	**Value:** 60
MEDIUM	**Value:** 40
NORMAL	**Value:** 30
SMALL	**Value:** 20
VERY_SMALL	**Value:** 15

33.9 Class Game

33.9.1 Methods

__init__(*self, map*)

create_empty_actions()

33.10 Class ResourceManager

Gerenciador de Recursos

33.10.1 Methods

__init__(*self, dir_root=*None)

Construtor

Parameters
    dir_root: Diretório raiz

Return Value
    Instância

get_font(*self, size=*30)

Obtém Fonte padrão

Parameters
    size: Tamanho

Return Value
    Fonte

get_font_by_name(*self, name, size*)

Obtém Fonte pelo nome

Parameters
    name: Nome

    size: Tamanho

Return Value
    Fonte

get_font_default_name(*self*)

Obtém Fonte padrão

Return Value
    Fonte

---

get_font_large(*self*, *name*)

Obtém Fonte Grande

Parameters
    name: Nome

Return Value
    Fonte

---

get_font_normal(*self*, *name*)

Obtém Fonte Normal

Parameters
    name: Nome

Return Value
    Fonte

---

get_font_path(*self*, *name*)

Obtém caminho da fonte

Parameters
    name: Nome

Return Value
    Fonte

---

get_font_small(*self*, *name*)

Obtém Fonte pequena

Parameters
    name: Nome

Return Value
    Fonte

---

get_font_very_small(*self*, *name*)

Obtém Fonte muito pequena

Parameters
    `name`: Nome

Return Value
    Fonte

---

get_image(*self*, *name*)

Obtém imagem

Parameters
    `name`: Nome

Return Value
    Imagem

---

get_scalled_image(*self*, *name*, *scale*)

---

init_mixer(*self*)

Inicia mixer

---

list_files(*self*, *dir*)

---

load(*self*, *dir_root*)

Carregar

Parameters
    `dir_root`: Diretório raiz

---

load_font(*self*)

Carregar fontes

---

load_image(*self*)

Carregar imagens

---

load_music(*self*)

Carregar efeitos sonoros

---

load_sound(*self*)

Carregar efeitos sonoros

play_music(*self*, *name*, *loops=*1)

Reproduz música

Parameters

    `name`: Nome

    `loops`: Quantas vezes repetir

---

play_sound(*self*, *name*, *loops=*1)

Reproduzir efeito sonoro

Parameters

    `name`: Nome

    `loops`: Repetição

33.10.2 Class Variables

Name	Description
fonts	**Value:** {}
images	**Value:** {}
musics	**Value:** {}
sounds	**Value:** {}

33.11 Class Ship

pygame-asteroids.sprite_collision.SpriteCollision ─┐

object ─┐

                                  pygame.sprite.Sprite ─┐

                                          pygame-asteroids.ship.Ship

Nave

33.11.1 Methods

__init__(*self, location, game_controller, *groups*)

Construtor

Parameters
    `location`:

    `game_controller`:

    `groups`:

Return Value
    Instância

Overrides: object.__init__

---

accelerate(*self, angle*)

---

accelerate_speed(*self, angle, x, y*)

Acelerar

Parameters
    `angle`: Ângulo

    `x`:   Posição X

    `y`:   Posição Y

---

shoot(*self*)

Atirar

---

get_center(*self*)

---

explode(*self, time*)

Explidir nave

Parameters
    `time`: Delta Time

---

left(*self, time*)

---

right(*self, time*)

---

up(*self*)

---

rotate(*self, angle*)

---

update(*self, time*)

Atualizar

Parameters
    time: Delta Time

Overrides: pygame.sprite.Sprite.update

Inherited from pygame-asteroids.sprite_collision.SpriteCollision(Section 33.12)

collision(), draw2(), get_class_name(), initrect(), rectupdate(), screencollision()

Inherited from pygame.sprite.Sprite

__repr__(), add(), add_internal(), alive(), groups(), kill(), remove(), remove_internal()

Inherited from object

__delattr__(), __format__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

33.11.2 Properties

Name	Description
Inherited from object	
__class__	

33.11.3 Class Variables

Name	Description
SPEED_ACCELERATE	Velociadade de aceleração **Value:** `0.025`
SPEED_ROTATE	Velocidade de rotação **Value:** `0.4`
TIME_LOAD	Tempo para carregar **Value:** `200.0`
TIME_RELOAD_SHOT	Tempo para recarregar **Value:** `100`

33.12   Class SpriteCollision

Known Subclasses: pygame-asteroids.asteroid.Asteroid,   pygame-asteroids.bullet.Bullet, pygame-asteroids.ship.Ship

Classe principal que controla sprite e aplicada a todos os objetos em movimento

33.12.1   Methods

---

initrect(*self*, *location*, *game_controller*)

Iniciliza retângulo

Parameters
     `location:`

     `game_controller:`

---

get_class_name(*self*)

Retorna nome da classe instanciada

Return Value
     Nome

---

screencollision(*self*, *time*)

Envolve a tela do jogo para objetos

Parameters
     `time:` Delta time

---

rectupdate(*self*)

---

define qual borda colide com o objeto de colisão / blitting

---

draw2(*self*, *screen*)

---

blit de imagens para corrigir os lugares quando perto cantos / lados.

Parameters
     `screen:` Tela

---

collision(*self, object*)

Detecta colisão

Parameters
     `object`: Objeto

Return Value
     Booleano

33.13   Class StarField

object ─┐

       pygame-asteroids.start_field.StarField

33.13.1   Methods

__init__(*self, screen*)

Create the starfield

Overrides: object.__init__

---

move(*self, start, end, direction*)

Correct for stars hitting the screen's borders

---

fix_stars_on_screen(*self, star*)

---

draw(*self*)

Inherited from object

     __delattr__(), __format__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

33.13.2   Properties

Name	Description
Inherited from object	
__class__	

33.13.3   Class Variables

Name	Description
NUM_STARS	**Value:** 30
WHITE	**Value:** (255, 255, 255)
LEFT	**Value:** 0
RIGHT	**Value:** 1
direction	**Value:** 0
stars	**Value:** []

33.14   Class TextInput

object ──┐
         └── **pygame-asteroids.virtual_keyboard.TextInput**

Handles the text input box and manages the cursor

33.14.1   Methods

__init__(*self, background, screen, font, text, x, y*)

x.__init__(...) initializes x; see help(type(x)) for signature

Overrides: object.__init__ extit(inherited documentation)

---

draw(*self*)

Draw the text input box

---

flashcursor(*self*)

Toggle visibility of the cursor

---

addcharatcursor(*self, letter*)

Add a character whereever the cursor is currently located

---

backspace(*self*)

Delete a character before the cursor position

---

deccursor(*self*)

Move the cursor one space left

256

inccursor(*self*)
Move the cursor one space right (but not beyond the end of the text)

drawcursor(*self*)
Draw the cursor

Inherited from object

__delattr__(), __format__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

33.14.2   Properties

Name	Description
Inherited from object	
__class__	

33.15   Class TouchButtons

33.15.1   Methods

__init__(*self, screen, button_width*=40)

create_touch_buttons(*self, button_width*)

draw(*self*)

detect_actions(*self, event, actions*)

33.16   Class VirtualKey

object ─┐
        pygame-asteroids.virtual_keyboard.VirtualKey

A single key for the VirtualKeyboard

33.16.1 Methods

__init__(*self, caption, x, y, w*=67, *h*=67)

x.__init__(...) initializes x; see help(type(x)) for signature

Overrides: object.__init__ extit(inherited documentation)

---
draw(*self, screen, background, shifted*=False, *forcedraw*=False)

Draw one key if it needs redrawing

Inherited from object

__delattr__(), __format__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

33.16.2 Properties

Name	Description
Inherited from object	
__class__	

33.17 Class VirtualKeyboard

object ─┐

       pygame-asteroids.virtual_keyboard.VirtualKeyboard

Implement a basic full screen virtual keyboard for touchscreens

33.17.1 Methods

---
run(*self, screen, font, text*='', *callback*=None)

---
not_null(*self, text*)

---
unselectall(*self, force*=False)

Force all the keys to be unselected Marks any that change as dirty to redraw

---

clickatmouse(*self*)

Check to see if the user is pressing down on a key and draw it selected

---

togglecaps(*self*)

Toggle uppercase / lowercase

---

selectatmouse(*self*)

User has clicked a key, let's use it

---

addkeys(*self*)

Adds the setup for the keys. This would be easy to modify for additional keys

The default start position places the keyboard slightly left of center by design so many people have issues with the right side of their touchscreens that I did this on purpose.

---

paintkeys(*self*)

Draw the keyboard (but only if they're dirty.)

---

clear(*self*)

Put the screen back to before we started

Inherited from object

__delattr__(), __format__(), __getattribute__(), __hash__(), __init__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

33.17.2 Properties

Name	Description
Inherited from object	
__class__	

33.18   Class error

object ⌐

exceptions.BaseException ⌐

exceptions.Exception ⌐

resource.error

33.18.1   Methods

Inherited from exceptions.Exception

__init__(), __new__()

Inherited from exceptions.BaseException

__delattr__(), __getattribute__(), __getitem__(), __getslice__(), __reduce__(), __repr__(), __setattr__(), __setstate__(), __str__(), __unicode__()

Inherited from object

__format__(), __hash__(), __reduce_ex__(), __sizeof__(), __subclasshook__()

33.18.2   Properties

Name	Description
Inherited from exceptions.BaseException	
args, message	
Inherited from object	
__class__	

33.19   Class struct_rusage

object ⌐

resource.struct_rusage

```
struct_rusage: Result from getrusage.
```

```
This object may be accessed either as a tuple of
    (utime,stime,maxrss,ixrss,idrss,isrss,minflt,majflt,
```

```
    nswap,inblock,oublock,msgsnd,msgrcv,nsignals,nvcsw,nivcsw)
or via the attributes ru_utime, ru_stime, ru_maxrss, and so on.
```

33.19.1 Methods

__add__(x, y)

x+y

__contains__(x, y)

y in x

__eq__(x, y)

x==y

__ge__(x, y)

x>=y

__getitem__(x, y)

x[y]

__getslice__(x, i, j)

x[i:j]

Use of negative indices is not supported.

__gt__(x, y)

x>y

__hash__(x)

hash(x)

Overrides: object.__hash__

__le__(x, y)

x<=y

__len__(*x*)

len(x)

__lt__(*x, y*)

x<y

__mul__(*x, n*)

x*n

__ne__(*x, y*)

x!=y

__new__(*T, S, ...*)

Return Value

   a new object with type S, a subtype of T

Overrides: object.__new__

__reduce__(*...*)

helper for pickle

Overrides: object.__reduce__ extit(inherited documentation)

__repr__(*x*)

repr(x)

Overrides: object.__repr__

__rmul__(*x, n*)

n*x

Inherited from object

__delattr__(), __format__(), __getattribute__(), __init__(), __reduce_ex__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

33.19.2 Properties

Name	Description
ru_idrss	unshared data size
ru_inblock	block input operations

Name	Description
ru_isrss	unshared stack size
ru_ixrss	shared memory size
ru_majflt	page faults requiring I/O
ru_maxrss	max. resident set size
ru_minflt	page faults not requiring I/O
ru_msgrcv	IPC messages received
ru_msgsnd	IPC messages sent
ru_nivcsw	involuntary context switches
ru_nsignals	signals received
ru_nswap	number of swap outs
ru_nvcsw	voluntary context switches
ru_oublock	block output operations
ru_stime	system time used
ru_utime	user time used
Inherited from object	
__class__	

33.19.3 Class Variables

Name	Description
n_fields	**Value:** 16
n_sequence_fields	**Value:** 16
n_unnamed_fields	**Value:** 0

34   Module pygame-asteroids.test_virtual_keyboard'

34.1   Functions

main()

34.2   Variables

Name	Description
__package__	**Value:** 'pygame-asteroids'

35   Module pygame-asteroids.tools'

This module contains the fundamental Control class. Also contained here are resource loading functions.

35.1   Functions

---

cursor_from_image(*image*, *size*, *hotspot*, *location*=`(0, 0)`, *flip*=`False`)

This function's return value is of the form accepted by pg.mouse.set_cursor() (passed using the *args syntax). The argument image is an already loaded image surface containing your desired cursor; size is a single integer corresponding to the width of the cursor (must be a multiple of 8); hotspot is a 2-tuple representing the exact point in your cursor that will represent the mouse position; location is a 2-tuple for where your cursor is located on the passed in image. Setting flip to True will create the cursor with colors opposite to the source image.

Color in image to color in cursor defaults: Black (0, 0, 0) —> Black White (255, 255, 255) —> White Cyan (0, 255, 255) —> Xor (only available on certain systems) Any Other Color ——→ Transparent

---

get_cell_coordinates(*rect*, *point*, *size*)

Find the cell of size, within rect, that point occupies.

---

get_rendered(*font*, *text*, *color*, *cache*)

Simple font renderer that caches render.

---

load_all_fonts(*directory*, *accept*=`('.ttf')`)

Create a dictionary of paths to font files in given directory if their extensions are in accept.

---

load_all_gfx(*directory*, *colorkey*=`(255, 0, 255)`, *accept*=`('.png', '.jpg', '.bmp')`)

Load all graphics with extensions in the accept argument. If alpha transparency is found in the image the image will be converted using convert_alpha(). If no alpha transparency is detected image will be converted using convert() and colorkey will be set to colorkey.

---

load_all_music(*directory*, *accept*=('.wav', '.mp3', '.ogg', '.mdi'))

Create a dictionary of paths to music files in given directory if their extensions are in accept.

load_all_sfx(*directory*, *accept*=('.wav', '.mp3', '.ogg', '.mdi'))

Load all sfx of extensions found in accept. Unfortunately it is common to need to set sfx volume on a one-by-one basis. This must be done manually if necessary.

rect_then_mask(*one*, *two*)

This is a callback function to be used with sprite group collision methods. It initially checks if two sprites have overlapping rectangles. If this is True, it will check if their masks collide and return the result. If the rectangles were not colliding, the mask check is not performed.

strip_coords_from_sheet(*sheet*, *coords*, *size*)

Strip specific coordinates from a sprite sheet.

strip_from_sheet(*sheet*, *start*, *size*, *columns*, *rows*=1)

Strips individual frames from a sprite sheet given a start location, sprite size, and number of columns and rows.

35.2   Variables

Name	Description
TIME_PER_UPDATE	**Value:** `16.0`
__package__	**Value:** `'pygame-asteroids'`

35.3   Class Anim

object ¬

  pygame-asteroids.tools.Anim

A class to simplify the act of adding animations to sprites.

35.3.1   Methods

__init__(*self, frames, fps, loops=-1*)

The argument frames is a list of frames in the correct order; fps is the frames per second of the animation; loops is the number of times the animation will loop (a value of -1 will loop indefinitely).

Overrides: object.__init__

---

get_next_frame(*self, now*)

Advance the frame if enough time has elapsed and the animation has not finished looping.

---

reset(*self*)

Set frame, timer, and loop status back to the initialized state.

Inherited from object

__delattr__(), __format__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

35.3.2   Properties

Name	Description
Inherited from object	
__class__	

35.4   Class Control

object —┐
        └ **pygame-asteroids.tools.Control**

Control class for entire project. Contains the game loop, and contains the event_loop which passes events to States as needed.

35.4.1 Methods

__init__(*self, caption, fps*=60.0)

x.__init__(...) initializes x; see help(type(x)) for signature

Overrides: object.__init__ extit(inherited documentation)

---

draw(*self, interpolate*)

---

event_loop(*self*)

Process all events and pass them down to the state_machine. The f5 key globally turns on/off the display of FPS in the caption

---

main(*self*)

Main loop for entire program. Uses a constant timestep.

---

show_fps(*self*)

Display the current FPS in the window handle if fps_visible is True.

---

toggle_show_fps(*self, key*)

Press f5 to turn on/off displaying the framerate in the caption.

---

update(*self*)

Updates the currently active state.

Inherited from object

__delattr__(), __format__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

35.4.2 Properties

Name	Description
Inherited from object	
__class__	

35.5   Class StateMachine

object ─┐
      pygame-asteroids.state_machine.StateMachine

Construtor.

35.5.1   Methods

__init__(*self*)

x.__init__(...) initializes x; see help(type(x)) for signature

Overrides: object.__init__ extit(inherited documentation)

---

draw(*self, surface, interpolate*)

---

flip_state(*self*)

When a State changes to done necessary startup and cleanup functions are called and the current State is changed.

---

get_event(*self, event*)

Pass events down to current State.

---

setup_states(*self, state_dict, start_state*)

Given a dictionary of states and a state to start in, creates the self.state_dict.

---

update(*self, keys, now*)

Checks if a state is done or has called for a game quit. State is flipped if neccessary and State.update is called.

Inherited from object

    __delattr__(), __format__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

35.5.2   Properties

Name	Description
Inherited from object	
__class__	

Name	Description

35.6   Class Timer

object ──┐
          └── **pygame-asteroids.tools.Timer**

A very simple timer for events that are not directly tied to animation.

35.6.1   Methods

__init__(*self, delay, ticks=-1*)

The delay is given in milliseconds; ticks is the number of ticks the timer will make before flipping self.done to True. Pass a value of -1 to bypass this.

Overrides: object.__init__

---
check_tick(*self, now*)

Returns true if a tick worth of time has passed.

Inherited from object

__delattr__(), __format__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

35.6.2   Properties

Name	Description
Inherited from object	
__class__	

36   Module pygame-asteroids.touch_buttons

36.1   Variables

Name	Description
RLIMIT_AS	**Value:** 5
RLIMIT_CORE	**Value:** 4
RLIMIT_CPU	**Value:** 0
RLIMIT_DATA	**Value:** 2
RLIMIT_FSIZE	**Value:** 1
RLIMIT_MEMLOCK	**Value:** 6
RLIMIT_NOFILE	**Value:** 8
RLIMIT_NPROC	**Value:** 7
RLIMIT_RSS	**Value:** 5
RLIMIT_STACK	**Value:** 3
RLIM_INFINITY	**Value:** 9223372036854775807
RUSAGE_CHILDREN	**Value:** -1
RUSAGE_SELF	**Value:** 0
__package__	**Value:** 'pygame-asteroids'

36.2   Class TouchButtons

36.2.1   Methods

__init__(*self, screen, button_width*=40)

create_touch_buttons(*self, button_width*)

draw(*self*)

detect_actions(*self, event, actions*)

37 Module pygame-asteroids.virtual_controller

37.1 Functions

---
is_event_type_an_input_down(*event*)

---
decode_joystick_arrow_to_keyboard_key(*joystick*)

---
vc_get_angle(*center*, *pos*)

Parameters
    pos: posição do mouse

Return Value
    ângulo em raidanos da posição do mouse em relação ao centro
informado. Controle Virtual é dividido em três partes. Para o eixo-y
: 1) above the center of the controller 2) below the center of the
controller 3) at the same height of the center of the controller If the
mouse point is above the center of the controller, than check for one
of three conditions: 1) x is to the right of the controller 2) x is to the
left of the controller 3) x is at the same point as the centerx of the
controller

37.2 Variables

Name	Description
__package__	**Value:** 'pygame-asteroids'

38   Module pygame-asteroids.virtual_keyboard

38.1   Variables

Name	Description
__package__	**Value:** 'pygame-asteroids'

38.2   Class TextInput

object ─┐

        pygame-asteroids.virtual_keyboard.TextInput

Handles the text input box and manages the cursor

38.2.1   Methods

__init__(*self, background, screen, font, text, x, y*)

x.__init__(...) initializes x; see help(type(x)) for signature

Overrides: object.__init__ extit(inherited documentation)

---

draw(*self*)

Draw the text input box

---

flashcursor(*self*)

Toggle visibility of the cursor

---

addcharatcursor(*self, letter*)

Add a character whereever the cursor is currently located

---

backspace(*self*)

Delete a character before the cursor position

---

deccursor(*self*)

Move the cursor one space left

---

inccursor(*self*)

Move the cursor one space right (but not beyond the end of the text)

---

drawcursor(*self*)

Draw the cursor

Inherited from object

__delattr__(), __format__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

38.2.2 Properties

Name	Description
Inherited from object	
__class__	

38.3 Class VirtualKey

object —┐

        pygame-asteroids.virtual_keyboard.VirtualKey

A single key for the VirtualKeyboard

38.3.1 Methods

__init__(*self, caption, x, y, w=*67*, h=*67)

x.__init__(...) initializes x; see help(type(x)) for signature

Overrides: object.__init__ extit(inherited documentation)

---

draw(*self, screen, background, shifted=*False*, forcedraw=*False)

Draw one key if it needs redrawing

Inherited from object

__delattr__(), __format__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

38.3.2 Properties

Name	Description
Inherited from object	
__class__	

38.4 Class VirtualKeyboard

object ─┐

pygame-asteroids.virtual_keyboard.VirtualKeyboard

Implement a basic full screen virtual keyboard for touchscreens

38.4.1 Methods

---

run(*self, screen, font, text=''*, *callback=*None)

---

not_null(*self, text*)

---

unselectall(*self, force=*False)

Force all the keys to be unselected Marks any that change as dirty to redraw

---

clickatmouse(*self*)

Check to see if the user is pressing down on a key and draw it selected

---

togglecaps(*self*)

Toggle uppercase / lowercase

---

selectatmouse(*self*)

User has clicked a key, let's use it

---

addkeys(*self*)

Adds the setup for the keys. This would be easy to modify for additional keys

The default start position places the keyboard slightly left of center by design so many people have issues with the right side of their touchscreens that I did this on purpose.

---

paintkeys(*self*)

Draw the keyboard (but only if they're dirty.)

clear(*self*)

Put the screen back to before we started

Inherited from object

__delattr__(), __format__(), __getattribute__(), __hash__(), __init__(), __new__(), __reduce__(),
__reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

38.4.2 Properties

Name	Description
Inherited from object	
__class__	

39   Module random

Random variable generators.

```
    integers
    --------
            uniform within range

    sequences
    ---------
            pick random element
            pick random sample
            generate random permutation

    distributions on the real line:
    -------------------------------
            uniform
            triangular
            normal (Gaussian)
            lognormal
            negative exponential
            gamma
            beta
            pareto
            Weibull

    distributions on the circle (angles 0 to 2pi)
    ---------------------------------------------
            circular uniform
            von Mises
```

General notes on the underlying Mersenne Twister core generator:

* The period is 2**19937-1.
* It is one of the most extensively tested generators in existence.
* Without a direct way to compute N steps forward, the semantics of
  jumpahead(n) are weakened to simply jump to another distant state and rely
  on the large period to avoid overlapping sequences.
* The random() method is implemented in C, executes in a single Python step,
  and is, therefore, threadsafe.

39.1   Functions

---

betavariate(*alpha, beta*)

Beta distribution.

Conditions on the parameters are alpha > 0 and beta > 0. Returned values range between 0 and 1.

---

choice(*seq*)

Choose a random element from a non-empty sequence.

---

expovariate(*lambd*)

Exponential distribution.

lambd is 1.0 divided by the desired mean. It should be nonzero. (The parameter would be called "lambda", but that is a reserved word in Python.) Returned values range from 0 to positive infinity if lambd is positive, and from negative infinity to 0 if lambd is negative.

---

gammavariate(*alpha, beta*)

```
Gamma distribution.   Not the gamma function!

Conditions on the parameters are alpha > 0 and beta > 0.

The probability distribution function is:

            x ** (alpha - 1) * math.exp(-x / beta)
  pdf(x) =  --------------------------------------
               math.gamma(alpha) * beta ** alpha
```

---

gauss(*mu, sigma*)

Gaussian distribution.

mu is the mean, and sigma is the standard deviation. This is slightly faster than the normalvariate() function.

Not thread-safe without a lock around calls.

---

getrandbits(*k*)

Generates a long int with k random bits.

Return Value
    x

---

getstate()

Return internal state; can be passed to setstate() later.

---

jumpahead(*n*)

Change the internal state to one that is likely far away from the current state. This method will not be in Py3.x, so it is better to simply reseed.

---

lognormvariate(*mu*, *sigma*)

Log normal distribution.

If you take the natural logarithm of this distribution, you'll get a normal distribution with mean mu and standard deviation sigma. mu can have any value, and sigma must be greater than zero.

---

normalvariate(*mu*, *sigma*)

Normal distribution.

mu is the mean, and sigma is the standard deviation.

---

paretovariate(*alpha*)

Pareto distribution. alpha is the shape parameter.

---

randint(*a*, *b*)

Return random integer in range [a, b], including both end points.

---

random()

Return Value
    x in the interval [0, 1).

randrange(*start*, *stop*=None, *step*=1, *_int*=<type 'int'>,
_maxwidth=9007199254740992)

Choose a random item from range(start, stop[, step]).

This fixes the problem with randint() which includes the endpoint; in Python this is usually not what you want.

---

sample(*population*, *k*)

Chooses k unique random elements from a population sequence.

Returns a new list containing elements from the population while leaving the original population unchanged. The resulting list is in selection order so that all sub-slices will also be valid random samples. This allows raffle winners (the sample) to be partitioned into grand prize and second place winners (the subslices).

Members of the population need not be hashable or unique. If the population contains repeats, then each occurrence is a possible selection in the sample.

To choose a sample in a range of integers, use xrange as an argument. This is especially fast and space efficient for sampling from a large population: sample(xrange(10000000), 60)

---

seed(*a*=None)

Initialize internal state from hashable object.

None or no argument seeds from current time or from an operating system specific randomness source if available.

If a is not None or an int or long, hash(a) is used instead.

---

setstate(*state*)

Restore internal state from object returned by getstate().

---

shuffle(*x*, *random*=None)

---

x, random=random.random -> shuffle list x in place; return None.

Optional arg random is a 0-argument function returning a random float in [0.0, 1.0); by default, the standard random.random.

---

triangular(*low*=0.0, *high*=1.0, *mode*=None)

Triangular distribution.

Continuous distribution bounded by given lower and upper limits, and having a given mode value in-between.

http://en.wikipedia.org/wiki/Triangular_distribution

---

uniform(*a*, *b*)

Get a random number in the range [a, b) or [a, b] depending on rounding.

---

vonmisesvariate(*mu*, *kappa*)

Circular data distribution.

mu is the mean angle, expressed in radians between 0 and 2*pi, and kappa is the concentration parameter, which must be greater than or equal to zero. If kappa is equal to zero, this distribution reduces to a uniform random angle over the range 0 to 2*pi.

---

weibullvariate(*alpha*, *beta*)

Weibull distribution.

alpha is the scale parameter and beta is the shape parameter.

39.2   Class Random

object ─┐

_random.Random ─┐
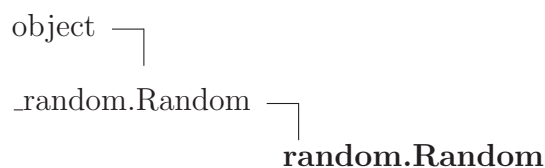
                         random.Random

Known Subclasses: random.SystemRandom, random.WichmannHill

Random number generator base class used by bound module functions.

Used to instantiate instances of Random to get generators that don't share state. Especially useful for multi-threaded programs, creating a different instance of Random for each thread, and using the jumpahead() method to ensure that the generated sequences seen by each thread don't overlap.

Class Random can also be subclassed if you want to use a different basic generator of your own devising: in that case, override the following methods: random(), seed(), getstate(), set-

state() and jumpahead(). Optionally, implement a getrandbits() method so that randrange() can cover arbitrarily large ranges.

39.2.1 Methods

__**getstate**__(*self*)

__**init**__(*self*, *x*=`None`)

Initialize an instance.

Optional argument x controls seeding, as for Random.seed().

Overrides: object.__init__

__**reduce**__(*self*)

helper for pickle

Overrides: object.__reduce__ extit(inherited documentation)

__**setstate**__(*self*, *state*)

---

betavariate(*self*, *alpha*, *beta*)

Beta distribution.

Conditions on the parameters are alpha > 0 and beta > 0. Returned values range between 0 and 1.

---

choice(*self*, *seq*)

Choose a random element from a non-empty sequence.

---

expovariate(*self*, *lambd*)

Exponential distribution.

lambd is 1.0 divided by the desired mean. It should be nonzero. (The parameter would be called "lambda", but that is a reserved word in Python.) Returned values range from 0 to positive infinity if lambd is positive, and from negative infinity to 0 if lambd is negative.

---

gammavariate(*self, alpha, beta*)

```
Gamma distribution.  Not the gamma function!

Conditions on the parameters are alpha > 0 and beta > 0.

The probability distribution function is:

          x ** (alpha - 1) * math.exp(-x / beta)
  pdf(x) =  ------------------------------------
              math.gamma(alpha) * beta ** alpha
```

---

gauss(*self, mu, sigma*)

Gaussian distribution.

mu is the mean, and sigma is the standard deviation. This is slightly faster than the normalvariate() function.

Not thread-safe without a lock around calls.

---

getstate(*self*)

Return internal state; can be passed to setstate() later.

Return Value
    tuple containing the current state.

Overrides: _random.Random.getstate

---

jumpahead(*self, n*)

Change the internal state to one that is likely far away from the current state. This method will not be in Py3.x, so it is better to simply reseed.

Return Value
    None

Overrides: _random.Random.jumpahead

---

lognormvariate(*self, mu, sigma*)

Log normal distribution.

If you take the natural logarithm of this distribution, you'll get a normal distribution with mean mu and standard deviation sigma. mu can have any value, and sigma must be greater than zero.

---

normalvariate(*self, mu, sigma*)

Normal distribution.

mu is the mean, and sigma is the standard deviation.

---

paretovariate(*self, alpha*)

Pareto distribution. alpha is the shape parameter.

---

randint(*self, a, b*)

Return random integer in range [a, b], including both end points.

---

randrange(*self, start, stop*=None, *step*=1, *_int*=<type 'int'>, *_maxwidth*=9007199254740992)

Choose a random item from range(start, stop[, step]).

This fixes the problem with randint() which includes the endpoint; in Python this is usually not what you want.

---

sample(*self, population, k*)

Chooses k unique random elements from a population sequence.

Returns a new list containing elements from the population while leaving the original population unchanged. The resulting list is in selection order so that all sub-slices will also be valid random samples. This allows raffle winners (the sample) to be partitioned into grand prize and second place winners (the subslices).

Members of the population need not be hashable or unique. If the population contains repeats, then each occurrence is a possible selection in the sample.

To choose a sample in a range of integers, use xrange as an argument. This is especially fast and space efficient for sampling from a large population: sample(xrange(10000000), 60)

---

seed(*self*, *a*=`None`)

Initialize internal state from hashable object.

None or no argument seeds from current time or from an operating system specific randomness source if available.

If a is not None or an int or long, hash(a) is used instead.

Return Value
> None

Overrides: _random.Random.seed

---

setstate(*self*, *state*)

Restore internal state from object returned by getstate().

Return Value
> None

Overrides: _random.Random.setstate

---

shuffle(*self*, *x*, *random*=`None`)

x, random=random.random -> shuffle list x in place; return None.

Optional arg random is a 0-argument function returning a random float in [0.0, 1.0); by default, the standard random.random.

---

triangular(*self*, *low*=`0.0`, *high*=`1.0`, *mode*=`None`)

Triangular distribution.

Continuous distribution bounded by given lower and upper limits, and having a given mode value in-between.

http://en.wikipedia.org/wiki/Triangular_distribution

---

uniform(*self*, *a*, *b*)

Get a random number in the range [a, b) or [a, b] depending on rounding.

---

vonmisesvariate(*self*, *mu*, *kappa*)

Circular data distribution.

mu is the mean angle, expressed in radians between 0 and 2*pi, and kappa is the concentration parameter, which must be greater than or equal to zero. If kappa is equal to zero, this distribution reduces to a uniform random angle over the range 0 to 2*pi.

---

weibullvariate(*self*, *alpha*, *beta*)

Weibull distribution.

alpha is the scale parameter and beta is the shape parameter.

Inherited from _random.Random

__getattribute__(), __new__(), getrandbits(), random()

Inherited from object

__delattr__(), __format__(), __hash__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

39.2.2 Properties

Name	Description
Inherited from object	
__class__	

39.2.3 Class Variables

Name	Description
VERSION	Value: 3

39.3 Class SystemRandom

object ─┐
_random.Random ─┐
    random.Random ─┐
        random.SystemRandom
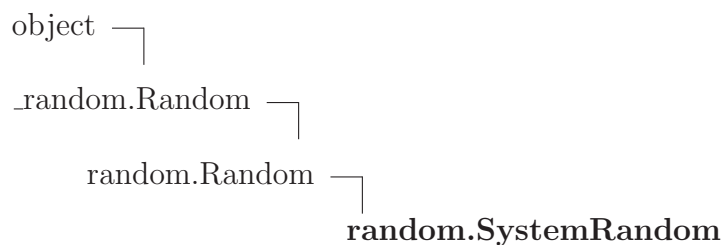
Alternate random number generator using sources provided
by the operating system (such as /dev/urandom on Unix or
CryptGenRandom on Windows).

 Not available on all systems (see os.urandom() for details).

39.3.1   Methods

---

getrandbits(*k*)

Generates a long int with k random bits.

Return Value
     x

Overrides: _random.Random.getrandbits

---

getstate(*self*, **args*, ***kwds*)

Method should not be called for a system random number generator.

Return Value
     None

Overrides: _random.Random.getstate

---

jumpahead(*self*, **args*, ***kwds*)

Stub method. Not used for a system random number generator.

Return Value
     None

Overrides: _random.Random.jumpahead

---

random(*self*)

Get the next random number in the range [0.0, 1.0).

Return Value
     x in the interval [0, 1).

Overrides: _random.Random.random

---

seed(*self*, **args*, ***kwds*)

Stub method. Not used for a system random number generator.

Return Value
     None

Overrides: _random.Random.seed

---

setstate(*self*, *args*, ***kwds*)

Method should not be called for a system random number generator.

Return Value
    None

Overrides: _random.Random.setstate

Inherited from random.Random(Section 39.2)

__getstate__(), __init__(), __reduce__(), __setstate__(), betavariate(), choice(), expovariate(), gammavariate(), gauss(), lognormvariate(), normalvariate(), paretovariate(), randint(), randrange(), sample(), shuffle(), triangular(), uniform(), vonmisesvariate(), weibullvariate()

Inherited from _random.Random

__getattribute__(), __new__()

Inherited from object

__delattr__(), __format__(), __hash__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

39.3.2   Properties

Name	Description
Inherited from object	
__class__	

39.3.3   Class Variables

Name	Description
Inherited from random.Random (Section 39.2)	
VERSION	

39.4   Class WichmannHill

object ┐
    _random.Random ┐
        random.Random ┐
                random.WichmannHill
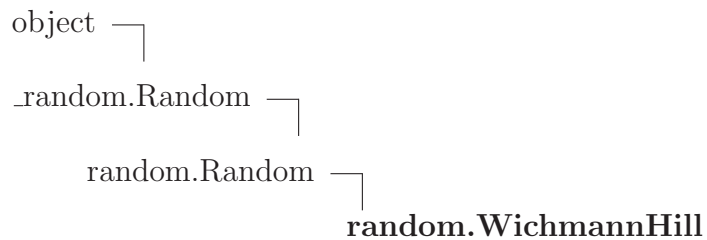
39.4.1   Methods

---
getstate(*self*)

Return internal state; can be passed to setstate() later.

Return Value
> tuple containing the current state.

Overrides: _random.Random.getstate

---
jumpahead(*self*, *n*)

```
Act as if n calls to random() were made, but quickly.

n is an int, greater than or equal to 0.

Example use:  If you have 2 threads and know that each will
consume no more than a million random numbers, create two Random
objects r1 and r2, then do
    r2.setstate(r1.getstate())
    r2.jumpahead(1000000)
Then r1 and r2 will use guaranteed-disjoint segments of the full
period.
```

Return Value
> None

Overrides: _random.Random.jumpahead

---
random(*self*)

Get the next random number in the range [0.0, 1.0).

Return Value
> x in the interval [0, 1).

Overrides: _random.Random.random

---

seed(*self*, *a*=None)

Initialize internal state from hashable object.

None or no argument seeds from current time or from an operating system specific randomness source if available.

If a is not None or an int or long, hash(a) is used instead.

If a is an int or long, a is used directly. Distinct values between 0 and 27814431486575L inclusive are guaranteed to yield distinct internal states (this guarantee is specific to the default Wichmann-Hill generator).

Return Value
    None

Overrides: _random.Random.seed

---

setstate(*self*, *state*)

Restore internal state from object returned by getstate().

Return Value
    None

Overrides: _random.Random.setstate

---

whseed(*self*, *a*=None)

Seed from hashable object's hash code.

None or no argument seeds from current time. It is not guaranteed that objects with distinct hash codes lead to distinct internal states.

This is obsolete, provided for compatibility with the seed routine used prior to Python 2.1. Use the .seed() method instead.

Inherited from random.Random(Section 39.2)

__getstate__(), __init__(), __reduce__(), __setstate__(), betavariate(), choice(), expovariate(), gammavariate(), gauss(), lognormvariate(), normalvariate(), paretovariate(), randint(), randrange(), sample(), shuffle(), triangular(), uniform(), vonmisesvariate(), weibullvariate()

Inherited from _random.Random

__getattribute__(), __new__(), getrandbits()

Inherited from object

__delattr__(), __format__(), __hash__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(),

__str__(), __subclasshook__()

39.4.2   Properties

Name	Description
Inherited from object	
__class__	

39.4.3   Class Variables

Name	Description
VERSION	**Value:** 1

40 Module state_machine

Generalização de uma máquina de estado. Utilizada para gerenciar o fluxo do programa

40.1 Variables

Name	Description
__package__	**Value:** `None`

40.2 Class StateMachine

object ┐

     state_machine.StateMachine

Construtor.

40.2.1 Methods

__init__(*self*)

x.__init__(...) initializes x; see help(type(x)) for signature

Overrides: object.__init__ extit(inherited documentation)

---

draw(*self*, *surface*, *interpolate*)

---

flip_state(*self*)

When a State changes to done necessary startup and cleanup functions are called and the current State is changed.

---

get_event(*self*, *event*)

Pass events down to current State.

---

setup_states(*self*, *state_dict*, *start_state*)

Given a dictionary of states and a state to start in, creates the self.state_dict.

---

update(*self*, *keys*, *now*)

Checks if a state is done or has called for a game quit. State is flipped if neccessary and State.update is called.

Inherited from object

__delattr__(), __format__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

40.2.2 Properties

Name	Description
Inherited from object	
__class__	

41   Module sys

This module provides access to some objects used or maintained by the interpreter and to functions that interact strongly with the interpreter.

Dynamic objects:

argv -- command line arguments; argv[0] is the script pathname if known
path -- module search path; path[0] is the script directory, else ''
modules -- dictionary of loaded modules

displayhook -- called to show results in an interactive session
excepthook -- called to handle any uncaught exception other than SystemExit
  To customize printing in an interactive session or to install a custom
  top-level exception handler, assign other functions to replace these.

exitfunc -- if sys.exitfunc exists, this routine is called when Python exits
  Assigning to sys.exitfunc is deprecated; use the atexit module instead.

stdin -- standard input file object; used by raw_input() and input()
stdout -- standard output file object; used by the print statement
stderr -- standard error object; used for error messages
  By assigning other file objects (or objects that behave like files)
  to these, it is possible to redirect all of the interpreter's I/O.

last_type -- type of last uncaught exception
last_value -- value of last uncaught exception
last_traceback -- traceback of last uncaught exception
  These three are only available in an interactive session after a
  traceback has been printed.

exc_type -- type of exception currently being handled
exc_value -- value of exception currently being handled
exc_traceback -- traceback of exception currently being handled
  The function exc_info() should be used instead of these three,
  because it is thread-safe.

Static objects:

float_info -- a dict with information about the float inplementation.
long_info -- a struct sequence with information about the long implementation.
maxint -- the largest supported integer (the smallest is -maxint-1)
maxsize -- the largest supported length of containers.
maxunicode -- the largest supported character

builtin_module_names -- tuple of module names built into this interpreter
version -- the version of this interpreter as a string
version_info -- version information as a named tuple
hexversion -- version information encoded as a single integer
copyright -- copyright notice pertaining to this interpreter
platform -- platform identifier
executable -- absolute path of the executable binary of the Python interpreter
prefix -- prefix used to find the Python library
exec_prefix -- prefix used to find the machine-specific Python library
float_repr_style -- string indicating the style of repr() output for floats
__stdin__ -- the original stdin; don't touch!
__stdout__ -- the original stdout; don't touch!
__stderr__ -- the original stderr; don't touch!
__displayhook__ -- the original displayhook; don't touch!
__excepthook__ -- the original excepthook; don't touch!

Functions:

displayhook() -- print an object to the screen, and save it in __builtin__._
excepthook() -- print an exception and its traceback to sys.stderr
exc_info() -- return thread-safe information about the current exception
exc_clear() -- clear the exception state for the current thread
exit() -- exit the interpreter by raising SystemExit
getdlopenflags() -- returns flags to be used for dlopen() calls
getprofile() -- get the global profiling function
getrefcount() -- return the reference count for an object (plus one :-)
getrecursionlimit() -- return the max recursion depth for the interpreter
getsizeof() -- return the size of an object in bytes
gettrace() -- get the global debug tracing function
setcheckinterval() -- control how often the interpreter checks for events
setdlopenflags() -- set the flags to be used for dlopen() calls
setprofile() -- set the global profiling function
setrecursionlimit() -- set the max recursion depth for the interpreter
settrace() -- set the global debug tracing function

41.1   Functions

__displayhook__(*object*)
Print an object to sys.stdout and also save it in __builtin__._
Return Value   None

__excepthook__(*exctype, value, traceback*)

Handle an exception by displaying it with a traceback on sys.stderr.

Return Value
    None

---

call_tracing(*func, args*)

Call func(*args), while tracing is enabled. The tracing state is saved, and restored afterwards. This is intended to be called from a debugger from a checkpoint, to recursively debug some other code.

Return Value
    object

---

callstats()

```
Return a tuple of function call statistics, if CALL_PROFILE was defined
when Python was built.  Otherwise, return None.

When enabled, this function returns detailed, implementation-specific
details about the number of function calls executed. The return value is
a 11-tuple where the entries in the tuple are counts of:
0. all function calls
1. calls to PyFunction_Type objects
2. PyFunction calls that do not create an argument tuple
3. PyFunction calls that do not create an argument tuple
   and bypass PyEval_EvalCodeEx()
4. PyMethod calls
5. PyMethod calls on bound methods
6. PyType calls
7. PyCFunction calls
8. generator calls
9. All other calls
10. Number of stack pops performed by call_function()
```

Return Value
    tuple of integers

---

displayhook(*object*)

Print an object to sys.stdout and also save it in __builtin__.

Return Value
    None

---

exc_clear()

Clear global information on the current exception. Subsequent calls to exc_info() will return (None,None,None) until another exception is raised in the current thread or the execution stack returns to a frame where another exception is being handled.

Return Value
      None

exc_info()

Return information about the most recent exception caught by an except clause in the current stack frame or in an older stack frame.

Return Value
      (type, value, traceback)

excepthook(*exctype*, *value*, *traceback*)

Handle an exception by displaying it with a traceback on sys.stderr.

Return Value
      None

exit(*status=...*)

Exit the interpreter by raising SystemExit(status). If the status is omitted or None, it defaults to zero (i.e., success). If the status is an integer, it will be used as the system exit status. If it is another kind of object, it will be printed and the system exit status will be one (i.e., failure).

getcheckinterval()

Return Value
      current check interval; see setcheckinterval().

getdefaultencoding()

Return the current default string encoding used by the Unicode implementation.

Return Value
      string

---

getdlopenflags()

Return the current value of the flags that are used for dlopen calls. The flag constants are defined in the ctypes and DLFCN modules.

Return Value
    int

---

getfilesystemencoding()

Return the encoding used to convert Unicode filenames in operating system filenames.

Return Value
    string

---

getprofile()

Return the profiling function set with sys.setprofile. See the profiler chapter in the library manual.

---

getrecursionlimit()

Return the current value of the recursion limit, the maximum depth of the Python interpreter stack. This limit prevents infinite recursion from causing an overflow of the C stack and crashing Python.

---

getrefcount(*object*)

Return the reference count of object. The count returned is generally one higher than you might expect, because it includes the (temporary) reference as an argument to getrefcount().

Return Value
    integer

---

getsizeof(*object*, *default*)

Return the size of object in bytes.

Return Value
    int

---

gettrace()

Return the global debug tracing function set with sys.settrace. See the debugger chapter in the library manual.

---

setcheckinterval(*n*)
Tell the Python interpreter to check for asynchronous events every n instructions. This also affects how often thread switches occur.

setdlopenflags(*n*)
Set the flags used by the interpreter for dlopen calls, such as when the interpreter loads extension modules. Among other things, this will enable a lazy resolving of symbols when importing a module, if called as sys.setdlopenflags(0). To share symbols across extension modules, call as sys.setdlopenflags(ctypes.RTLD_GLOBAL). Symbolic names for the flag modules can be either found in the ctypes module, or in the DLFCN module. If DLFCN is not available, it can be generated from /usr/include/dlfcn.h using the h2py script.
Return Value   None

setprofile(*function*)
Set the profiling function. It will be called on each function call and return. See the profiler chapter in the library manual.

setrecursionlimit(*n*)
Set the maximum depth of the Python interpreter stack to n. This limit prevents infinite recursion from causing an overflow of the C stack and crashing Python. The highest possible limit is platform- dependent.

settrace(*function*)
Set the global debug tracing function. It will be called on each function call. See the debugger chapter in the library manual.

41.2   Variables

Name	Description
__package__	**Value:** `None`
__stderr__	**Value:** `<open file '<stderr>', mode 'w' at 0x1002931e0>`
__stdin__	**Value:** `<open file '<stdin>', mode 'r' at 0x1002930c0>`
__stdout__	**Value:** `<open file '<stdout>', mode 'w' at 0x100293150>`
api_version	**Value:** `1013`

299

Name	Description
argv	**Value:** `['/Library/Frameworks/Python.framework/Versions/2.7/bin/e.`
builtin_module_names	**Value:** `('__builtin__', '__main__', '_ast', '_codecs', '_sre', '_...`
byteorder	**Value:** `'little'`
copyright	**Value:** `'Copyright (c) 2001-2014 Python Software Foundation.\nAll...`
dont_write_bytecode	**Value:** `False`
exc_type	**Value:** `None`
exec_prefix	**Value:** `'/Library/Frameworks/Python.framework/Versions/2.7'`
executable	**Value:** `'/Library/Frameworks/Python.framework/Versions/2.7/Resour.`
flags	**Value:** `sys.flags(debug=0, py3k_warning=0, division_warning=0, di...`
float_info	**Value:** `sys.float_info(max=1.7976931348623157e+308, max_exp=1024,...`
float_repr_style	**Value:** `'short'`
hexversion	**Value:** `34015728`
long_info	**Value:** `sys.long_info(bits_per_digit=30, sizeof_digit=4)`
maxint	**Value:** `9223372036854775807`
maxsize	**Value:** `9223372036854775807`
maxunicode	**Value:** `65535`
meta_path	**Value:** `[]`
modules	**Value:** `{'ConfigParser': <module 'ConfigParser' from '/Library/Fr...`
path	**Value:** `['/Users/tux/git', '/Library/Frameworks/Python.framework/...`
path_hooks	**Value:** `[<type 'zipimport.zipimporter'>]`
path_importer_cache	**Value:** `{'': None, '/Library/Frameworks/Python.framework/Versions...`
platform	**Value:** `'darwin'`
prefix	**Value:** `'/Library/Frameworks/Python.framework/Versions/2.7'`
py3kwarning	**Value:** `False`
stderr	**Value:** `<open file '<stderr>', mode 'w' at 0x1002931e0>`
stdin	**Value:** `<open file '<stdin>', mode 'r' at 0x1002930c0>`

Name	Description
stdout	**Value:** `<open file '<stdout>', mode 'w' at 0x100293150>`
subversion	**Value:** `('CPython', '', '')`
version	**Value:** `'2.7.9 (v2.7.9:648dcafa7e5f, Dec 10 2014, 10:10:46) \n[GC...`
version_info	**Value:** `sys.version_info(major=2, minor=7, micro=9, releaselevel=...`
warnoptions	**Value:** `[]`

42   Module time

This module provides various functions to manipulate time values.

There are two standard representations of time. One is the number
of seconds since the Epoch, in UTC (a.k.a. GMT). It may be an integer
or a floating point number (to represent fractions of seconds).
The Epoch is system-defined; on Unix, it is generally January 1st, 1970.
The actual value can be retrieved by calling gmtime(0).

The other representation is a tuple of 9 integers giving local time.
The tuple items are:
  year (four digits, e.g. 1998)
  month (1-12)
  day (1-31)
  hours (0-23)
  minutes (0-59)
  seconds (0-59)
  weekday (0-6, Monday is 0)
  Julian day (day in the year, 1-366)
  DST (Daylight Savings Time) flag (-1, 0 or 1)
If the DST flag is 0, the time is given in the regular time zone;
if it is 1, the time is given in the DST time zone;
if it is -1, mktime() should guess based on the date and time.

Variables:

timezone -- difference in seconds between UTC and local standard time
altzone -- difference in seconds between UTC and local DST time
daylight -- whether local time should reflect DST
tzname -- tuple of (standard time zone name, DST time zone name)

Functions:

time() -- return current time in seconds since the Epoch as a float
clock() -- return CPU time since process start as a float
sleep() -- delay for a number of seconds given as a float
gmtime() -- convert seconds since Epoch to UTC tuple
localtime() -- convert seconds since Epoch to local time tuple
asctime() -- convert time tuple to string
ctime() -- convert time in seconds to string
mktime() -- convert local time tuple to seconds since Epoch
strftime() -- convert time tuple to string according to format specification
strptime() -- parse string to time tuple according to format specification

```
tzset() -- change the local timezone
```

42.1   Functions

---

asctime(*tuple*=. . .)

Convert a time tuple to a string, e.g. 'Sat Jun 06 16:26:11 1998'. When the time tuple is not present, current time as returned by localtime() is used.

Return Value
> string

---

clock()

Return the CPU time or real time since the start of the process or since the first call to clock(). This has as much precision as the system records.

Return Value
> floating point number

---

ctime(*seconds*)

Convert a time in seconds since the Epoch to a string in local time. This is equivalent to asctime(localtime(seconds)). When the time tuple is not present, current time as returned by localtime() is used.

Return Value
> string

---

gmtime(*seconds*=. . .)

                           `tm_sec, tm_wday, tm_yday, tm_isdst)`

Convert seconds since the Epoch to a time tuple expressing UTC (a.k.a. GMT). When 'seconds' is not passed in, convert the current time instead.

Return Value
> (tm_year, tm_mon, tm_mday, tm_hour, tm_min,

---

localtime(*seconds=...*)

```
                          tm_sec,tm_wday,tm_yday,tm_isdst)
```

Convert seconds since the Epoch to a time tuple expressing local time.
When 'seconds' is not passed in, convert the current time instead.

Return Value
> (tm_year,tm_mon,tm_mday,tm_hour,tm_min,

---

mktime(*tuple*)

Convert a time tuple in local time to seconds since the Epoch.

Return Value
> floating point number

---

sleep(*seconds*)

Delay execution for a given number of seconds. The argument may be a
floating point number for subsecond precision.

---

strftime(*format, tuple=...*)

Convert a time tuple to a string according to a format specification. See the
library reference manual for formatting codes. When the time tuple is not
present, current time as returned by localtime() is used.

Return Value
> string

---

strptime(*string, format*)

Parse a string to a time tuple according to a format specification. See the
library reference manual for formatting codes (same as strftime()).

Return Value
> struct_time

---

time()

Return the current time in seconds since the Epoch. Fractions of a second may
be present if the system clock provides them.

Return Value
> floating point number

---

tzset()

Initialize, or reinitialize, the local timezone to the value stored in os.environ['TZ']. The TZ environment variable should be specified in standard Unix timezone format as documented in the tzset man page (eg. 'US/Eastern', 'Europe/Amsterdam'). Unknown timezones will silently fall back to UTC. If the TZ environment variable is not set, the local timezone is set to the systems best guess of wallclock time. Changing the TZ environment variable without calling tzset *may* change the local timezone used by methods such as localtime, but this behaviour should not be relied on.

42.2 Variables

Name	Description
__package__	**Value:** `None`
accept2dyear	**Value:** `1`
altzone	**Value:** `7200`
daylight	**Value:** `1`
timezone	**Value:** `10800`
tzname	**Value:** `('BRT', 'BRST')`

42.3 Class struct_time

object ──┐
       └── **time.struct_time**

The time value as returned by gmtime(), localtime(), and strptime(), and accepted by asctime(), mktime() and strftime(). May be considered as a sequence of 9 integers.

Note that several fields' values are not the same as those defined by the C language standard for struct tm. For example, the value of the field tm_year is the actual year, not year - 1900. See individual fields' descriptions for details.

42.3.1 Methods

__add__(*x, y*)

x+y

__contains__(*x, y*)

y in x

__eq__(*x*, *y*)

x==y

__ge__(*x*, *y*)

x>=y

__getitem__(*x*, *y*)

x[y]

__getslice__(*x*, *i*, *j*)

x[i:j]

Use of negative indices is not supported.

__gt__(*x*, *y*)

x>y

__hash__(*x*)

hash(x)

Overrides: object.__hash__

__le__(*x*, *y*)

x<=y

__len__(*x*)

len(x)

__lt__(*x*, *y*)

x<y

__mul__(*x*, *n*)

x*n

__ne__(*x*, *y*)

x!=y

__new__(*T, S, ...*)

Return Value
    a new object with type S, a subtype of T

Overrides: object.__new__

__reduce__(*...*)

helper for pickle

Overrides: object.__reduce__ extit(inherited documentation)

__repr__(*x*)

repr(x)

Overrides: object.__repr__

__rmul__(*x, n*)

n*x

Inherited from object

__delattr__(), __format__(), __getattribute__(), __init__(), __reduce_ex__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

42.3.2 Properties

Name	Description
tm_hour	hours, range [0, 23]
tm_isdst	1 if summer time is in effect, 0 if not, and -1 if unknown
tm_mday	day of month, range [1, 31]
tm_min	minutes, range [0, 59]
tm_mon	month of year, range [1, 12]
tm_sec	seconds, range [0, 61])
tm_wday	day of week, range [0, 6], Monday is 0
tm_yday	day of year, range [1, 366]
tm_year	year, for example, 1993
Inherited from object	
__class__	

42.3.3 Class Variables

Name	Description
n_fields	**Value:** 9
n_sequence_fields	**Value:** 9
n_unnamed_fields	**Value:** 0

Index