BlendDataWorlds(bpy_struct)

base class — `bpy_struct`

class bpy.types.**BlendDataWorlds(bpy_struct)**

Collection of worlds

new(name)

Add a new world to the main database

PARAMETERS:

name (*string, (never None)*) – New name for the data-block

RETURNS:

New world data-block

RETURN TYPE:

`World`

remove(world, *, do_unlink=True, do_id_user=True, do_ui_user=True)

Remove a world from the current blendfile

PARAMETERS:

- **world** (`World`, (never None)) – World to remove
- **do_unlink** (*boolean, (optional)*) – Unlink all usages of this world before deleting it
- **do_id_user** (*boolean, (optional)*) – Decrement user counter of all datablocks used by this world
- **do_ui_user** (*boolean, (optional)*) – Make sure interface does not reference this world

tag(value)

tag

PARAMETERS:

value (*boolean*) – Value

classmethod bl_rna_get_subclass(id, default=None)

PARAMETERS:

id (*str*) – The RNA type identifier.

RETURNS:

The RNA type or default when not found.

RETURN TYPE:

`bpy.types.Struct` subclass

classmethod bl_rna_get_subclass_py(id, default=None)

PARAMETERS:

id (*str*) – The RNA type identifier.

RETURNS:

The class or default when not found.

RETURN TYPE:

type

Inherited Properties

- `bpy_struct.id_data`

Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.id_properties_clear`
- `bpy_struct.id_properties_ensure`
- `bpy_struct.id_properties_ui`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_overridable_library`
- `bpy_struct.is_property_readonly`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.pop`
- `bpy_struct.property_overridable_library_set`
- `bpy_struct.property_unset`
- `bpy_struct.type_recast`
- `bpy_struct.values`

References

- `BlendData.worlds`