# Cheat Sheet viiapackage

Commonly used functions in the modelscript
(Date: 21/03/2024, package version: 70.3.0)

**1: Project, Grids and Layers**

| Function | Usage | Example |
|---|---|---|
| viia_create_project() | Sets up python Project (model) | project = viia_create_project( '1019A', object_part = 'Gehele object', version_nr = 1 ) |
| viia_create_orthogonal_grid() | Sets up Grid | grid_1 = project.viia_create_orthogonal_grid( [ [0, 1.4, 2.5], [0, 3.2] ] ) |
| viia_create_levels() | Adds Levels to a Grid | project.viia_create_levels( [-0.5, 0, 2.7, 5.0, 7.2], grid = grid_1 ) |
| viia_create_diagonal_gridlines_points_vector() | Adds diagonal gridlines to existing Grid | project.viia_create_diagonal_gridlines_points_vector(points = [ [1.0, 3.0] ], vector = [1.0, 2.0] , grid = grid_1) |

**2: View model**

| Function | Usage | Example |
|---|---|---|
| viia_plot_grid() | Plots all Grids and associated Levels | project.viia_plot_grid() |
| viia_plot_layers() | Plot some or all parts of the model (in 2D or 3D) | project.viia_plot_layers(layers=['N0', 'N1'], grid=grid_1 ,view = '3D', show=True) |
| viia_create_model_plots() | Makes model plots to be used in the model summary (Appendix C3) | project.viia_create_model_plots() |
| viia_create_model_plots_appendix() | Creates the model summary appendix, using the model plots | project. viia_create_model_plots_appendix() |
| viia_plot_psse_nsce() | Create plot of PSSE and NSCE decomposition | project. viia_plot_psse_nsce(show=True) |

**3: Basic Geometry (on a Grid)**

| Function | Usage | Example |
|---|---|---|
| viia_create_walls_on_grid() | Creates a set of walls along gridlines | n0_wall_list = [ [ 'D1', 'D9', 'MW-KLEI>1945', 210 ],<br>[ [5.0, 1.0], [2.0, 1.0] , 'MW-KLEI<1945', 210 ] ]<br>n0_walls = project. viia_create_walls_on_grid( layer='N0',<br>z_coordinate_bottom=0.0, z_coordinate_top=3.0, wall_list = n0_wall_list)) |
| viia_create_floors_on_grid() | Creates a set of floors along gridlines | n1_floor_list = [<br>['A1', 'B1', 'B2', 'A2', 'LIN-HBV-PLANKEN-0.03-0.040-0.150-0.600', 0, [0, 1, 0] ],<br>[ [0,0], [7.5, 0], [7.5,7.5], [0,7.5], 'LIN-BETON', 100, [1,0,0] ] ]<br>n1_floors = project.viia_create_floors_on_grid('N1', 3.0, n1_floor_list) |
| viia_beams_on_grid() | Creates a set of beams along gridlines | n2_beam_list = [ [ 'D1', 'D9', 3.0, 'LIN-STAAL-S355', 'HEA180' ],<br>[[0, 5.3], [7.5, 5.3], 4.6, 'LIN-HOUT', '100x100' ] ] )<br>n2_beams = project. viia_create_beams_on_grid(layer='N0',<br>beam_list= n2_beam_list ) |
| viia_create_columns_on_grid() | Creates a set of columns along gridlines | n0_column_list = [ [ 'D1', 0.0, 3.0, 'LIN-BETON', '210x210'],<br>[ [4.86, 6.01], 0.0, 3.0, 'LIN-HOUT', '50x100' ] ]<br>n0_columns = project. viia_create_columns_on_grid(layer = 'N0',<br>column_list = n0_column_list, is_truss = False) |
| viia_beams_on_grid_in_shape() | Creates a set of beams along gridlines in a shape (handy for roof beams in an angled roof) | gr_roof_beams = project.viia_create_orthogonal_grid( [ [0, 1.4, 2.5], [0, 3.2] ] )<br>roof_15 = project.viia_get( 'roofs', id = 15 )<br>n3_roof_beams = project. viia_create_beams_on_grid_in_shape( layer = 'N3',<br>grid = gr_roof_beams, gridlines = [ 'A', 'B', 'C' ], shape = roof_15,<br>material = 'LIN-HOUT-C14', geometry = 'D200', is_roof_beam = True ) |
| viia_create_collar_ties_on_grid() | Creates a set of collar ties ( horizontal tension elements in roofs) along gridlines | gr_roof_beams = project.viia_create_orthogonal_grid( [ [0, 1.4, 2.5], [0, 3.2] ] )<br>project.viia_create_levels( [2.7, 5.0, 7.2], grid = gr_roof_beams )<br>roof_list = [ project.viia_get( 'roofs', id = 15 ), project.viia_get( 'roofs', id = 16 ) ]<br>n3_collar_ties = project. viia_create_beams_on_grid_in_shape( layer = 'N3',<br>grid = gr_roof_beams, gridlines = [ 'A', 'B', 'C' ], shapes = roof_list,<br>beam_height = 'L2', material = 'LIN-HOUT-C14', geometry = 'D200',<br>is_truss = False, is_roof_beam = True ) |

**4: Basic Geometry (separate elements)**

| Function | Usage | Example |
|---|---|---|
| viia_create_beam() | Creates a single beam | edge_beam_balcony = project.viia_create_beam( name = 'N1',<br>material = 'LIN-BETON', geometry = '50x200',<br>points = [ [0.0, 5.0, 3.1], [2.0, 5.0, 3.1] ], is_roof_beam = False ) |
| viia_create_column() | Creates a single column | column_staircase = project.viia_create_column( name = 'N0',<br>material = 'LIN-BETON', geometry = '300x300',<br>points = [ [0.0, 5.0, 0.0], [0.0, 5.0, 3.1] ] ) |
| viia_create_floor() | Creates a single floor<br>( note the brackets in the *points* argument ) | birdhouse_floor = project.viia_create_floor(name = 'N0',<br>material = 'LIN-HOUT', geometry = '40',<br>points = [ [ [0, 1.0, 3.1], [2.0, 1.0, 3.1], [2.0, 5.0, 3.1], [0.0, 5.0, 3.1] ] ] ) |
| viia_create_roof() | Creates a single roof<br>( note the brackets in the *points* argument ) | terrace_roof = project.viia_create_roof(name = 'N3',<br>material = 'LIN-COSTEEL', geometry = '0.5',<br>points = [ [ [0, 1.0, 5.1], [2.0, 1.0, 5.1], [2.0, 5.0, 5.1], [0.0, 5.0, 5.1] ] ] ) |
| viia_create_wall() | Creates a single wall<br>( note the brackets in the *points* argument ) | game_room_wall = project.viia_create_wall(name = 'N2',<br>material = 'MW-KZS>1960', geometry = '100',<br>points = [ [ [0, 1.0, 3.1], [5.0, 1.0, 3.1], [5.0, 1.0, 5.1], [0.0, 1.0, 5.1] ] ] ) |

**5: Foundations**

| Function | Usage | Example |
|---|---|---|
| viia_create_foundation_walls_and_strips() | Adds walls and fstrips under selected walls | project.viia_create_foundation_walls_and_strips(walls = n0_walls,<br>foundation_depth = -1.0, strip_material = 'LIN-BETON', strip_width = 300,<br>strip_thickness = 200)<br><br>OR<br><br>project.viia_create_foundation_walls_and_strips(walls=n0_walls,<br>foundation_depth=-0.8,<br>stepped_foundation_strip_extensions=[120, 50, 60],<br>stepped_foundation_strip_heights=[60, 75, 40]) |
| viia_create_foundation_columns_and_strips() | Adds columns and fstrips under selected columns | project.viia_create_foundation_columns_and_strips(columns = n0_columns,<br>foundation_depth = -2.0, strip_material = 'LIN-BETON', strip_width = 300,<br>strip_thickness = 200 ) |
| viia_create_fstrip() | Creates one fstrip | project.viia_create_fstrip(name = 'F', material = 'LIN-BETON', geometry = '100',<br>points = [ [ [0.0, -0.5, -1.0], [0.0, 0.5, -1.0], [3.0, 0.5, -1.0], [3.0, -0.5, -1.0] ] ] ) |
| viia_adjust_fstrips() | Adjusts the contour of fstrips to line up | adjustable_fstrips = [ project.viia_get('fstrips', id = i) for i in [1, 2, 3, 4, 8] ]<br>project.viia_adjust_fstrips(adjustable_fstrips) |

**6: Edit Geometry**

| Function | Usage | Example |
|---|---|---|
| viia_get() | Search object in project | wall_1 = project.viia_get('walls', id=1) |
| viia_create_cavity_wall_ties_from_list() | Creates cavity walls and wall ties for a list of walls | project.viia_create_cavity_wall_ties_from_list(<br>cavity_wall_list = [ 'wall_1', 'wall_2' ], cavity = 0.1, ties_per_m2 = 4 ) |
| viia_connect_cavity_walls() | Connects 2 cavity walls | project.viia_connect_cavity_walls(wall_1, game_room_wall) |
| viia_create_wall_ties_on_line() | Creates a line of cavity wall ties | wall_1 = project.viia_get('walls', id=1)<br>wall_9001 = project.viia_get('walls', id=9001)<br>project.viia_create_wall_ties_on_line(wall_1, wall_9001,<br>inner_line = [ [0.0, 2.0, 2.2], [0.0, 5.0, 2.2] ],<br>outer_line = [ [0.2, 2.0, 2.2], [0.2, 5.0, 2.2] ] ) |
| viia_move_cavity_wall_tie() | Moves a cavity wall tie | tie_24 = project.viia_get_connection('springs', id = 24)<br>project.viia_move_cavity_wall_tie(tie_24, new_coordinates = [1.0, 2.0, 3.0] ) |
| viia_remove_cavity_wall_tie() | Remove a cavity wall tie | project.viia_remove_cavity_wall_tie(tie_24) |
| viia_add_openings() | Adds openings to walls and angled roofs | n2_opening_list = [<br>[ 'wall_6', [ [0.5, 2.0, 0.0, 2.2], [1.5, 2.0, 0.0, 2.2], [0.5, 2.0, 2.0, 2.2] ] ],<br>[ 'roof_8', [ [0.5, 2.0, 0.0, 2.2], [1.5, 2.0, 0.0, 2.2], [0.5, 2.0, 2.0, 2.2] ] ] ]<br>project.viia_add_openings(n2_opening_list) |
| viia_add_floor_openings() | Adds openings to floors and flat roofs | birdhouse_fl_opening_list = [ [ 1.0, 2.0, 3.0, 4.5 ], [ 2.2, 4.6, 7.2, 3.5 ] ]<br>project.viia_add_floor_openings(floor = birdhouse_floor,<br>opening_list = birdhouse_fl_opening_list) |
| viia_create_all_lintels() | Adds lintels to all openings in all walls | project.viia_create_all_lintels() |
| viia_create_lintel() | Adds one lintel to one lintel in a wall | project.viia_create_lintel(wall = game_room_wall, opening_nr = 2) |
| viia_split_surface() | Split or cut a surface shape | project.viia_split_surface(target = game_room_wall, tool = birdhouse_floor,<br>keep='largest') |
| viia_truss() | Transforms a line shape into a truss shape | project.viia_truss(column_staircase) |
| viia_remove_shape() | Removes shape and relations from project | project.viia_remove_shape(edge_beam_balcony) |

**7: Loads**

| Function | Usage | Example |
|---|---|---|
| viia_create_mass_all_openings() | Creates linemasses above and below all openings in walls and angled roofs | project.viia_create_mass_all_openings() |
| viia_create_mass_all__horizontal_surface_openings() | Creates linemasses around all openings in floors and flat roofs (not always required) | project.viia_create_mass_all_horizontal_surface_openings() |
| viia_create_linemass() | Adds a linemass to the model (recommended to also specify host) | floor_24 = project.viia_get('floors', id = 24)<br>linemass_staircase_1 = project.viia_create_linemass(layer = 'N2',<br>points = [ [1.0,2.0, 1.0], [3.0, 2.0, 1.0] ], value = 80, host = floor_24 ) |
| viia_create_linemass_beam() | Adds extra load as increased density to line shape (beam, column, lintel) | project.viia_create_linemass_beam(edge_beam_balcony, 1000) |
| viia_create_surface_mass_floor() | Adds extra (resting) load to floor or roof, adhering to BoD | project.viia_create_surface_mass_floor(floor=terrace_roof,<br>resting_load=['SlopeRoofLoad']) |
| viia_create_surface_mass_wall() | Adds extra (resting) load to wall, as increased density | project.viia_create_surface_mass_wall(wall=game_room_wall,<br>extra_load = 1000 ) |
| viia_adjust_mass_of_surface() | Adds custom resting load to surface, not adhering to BoD | project.viia_adjust_mass_of_surface(300, terrace_roof) |
| viia_create_loads() | Adds extra (imposed) load to shape(s) | loaded_floors = [<br>floor for floor in project.collections.floors if floor.id in [1, 2]]<br>project.viia_create_loads(load='Imposed load', category='AFloor',<br>connecting_shapes=loaded_floors) |

## 8: Handy functions from rhdhv_fem (need to be imported separately)

| Function | Usage | Example |
|---|---|---|
| fem_point_to_plane() | Calculates (and returns) a closest point on the plane from a reference point | from rhdhv_fem.fem_math import fem_point_to_plane<br>point = [2.0, 3.0, 2.0]<br>plane = [ [0.0, 0.0, 0.0], [1.0, 0.0, 0.0], [0.0, 1.0, 0.0] ]<br>point_on_plane = fem_point_to_plane(point, plane) |
| And a lot more, refer to https://datafusrfem.azurewebsites.net/ | | |

## 9: Finalize model

| Function | Usage | Example |
|---|---|---|
| viia_connect_all_shapes() | Connects (or merges) shapes that are the same shape, intersect, share nodes or share lines | project.viia_connect_all_shapes() |
| viia_create_all_datas() | Adds relevant data to shapes (for example about meshing, reinforcements or integration points) | project.viia_create_all_datas() |
| viia_write_dump() | Makes a .json format file of the project (to be used in further scripts) | project.viia_write_dump() |