varname def varname(frame: int=1, ignore: Optional[IgnoreType]=None, multi_vars:... node = get_node(frame + 1, ignore, raise_exc=raise_exc) /calls 1f not node: raise not node (not not node) if raise_exc: node = lookfor_parent_assign(node) ,'calls (not raise_exc) raise_exc lookfor_parent_assign raise VarnameRetrievingError('Unable to retrieve t...') if not node: not node (not not node) if isinstance(node, ast.AnnAssign): if raise_exc: (not raise_exc) __calls (isinstance(node, ast.AnnAssign) (not isinstance(node, ast.AnnAssign)) raise_exc raise VarnameRetrievingError('Failed to retrieve t...') isinstance target = node.target if len(node.targets) > 1: _____calls len(node.targets) > 1 warnings.warn('Multiple targets in ...'
, MultiTargetAssignmentWarning) (len(node.targets) <= 1) ∕calls target = node.targets[0] warnings.warn names = node_name(target) -- calls if not isinstance(names, tuple): (not isinstance(names, tuple)) names = names, (not not isinstance(names, tuple)) if multi_vars: if len(names) > 1: return names len(names) > 1 $(len(names) \ll 1)$ raise ImproperUseError(
 f'Expect a single vari...') return names[0]

KEY

return

default

input

call