

Yugo

This smart contract creates the ERC20 Yugo token.
Only the YugoManager contract can act on this contract.

Functions

constructor

Name	Type	Description
_manager	address	

Returns:
No parameters

DEFAULT_ADMIN_ROLE

****Add Documentation for the method here****

No parameters
Returns:

Name	Type	Description
	bytes32	

DOMAIN_SEPARATOR

See {IERC20Permit-DOMAIN_SEPARATOR}.

No parameters
Returns:

Name	Type	Description
	bytes32	

MINTER_ROLE

****Add Documentation for the method here****

No parameters
Returns:

Name	Type	Description
	bytes32	

PAUSER_ROLE

****Add Documentation for the method here****

No parameters
Returns:

Name	Type	Description
	bytes32	

allowance

See {IERC20-allowance}.

Name	Type	Description
owner	address	
spender	address	

Returns:

Name	Type	Description
	uint256	

approve

See {IERC20-approve}. Requirements: - `spender` cannot be the zero address.

Name	Type	Description
spender	address	
amount	uint256	

Returns:

Name	Type	Description
	bool	

balanceOf

See {IERC20-balanceOf}.

Name	Type	Description
account	address	

Returns:

Name	Type	Description
	uint256	

burn

Destroys `amount` tokens from the caller. See {ERC20-_burn}.

Name	Type	Description
amount	uint256	

Returns:

No parameters

burnFrom

Destroys `amount` tokens from `account`, deducting from the caller's allowance. See {ERC20-_burn} and {ERC20-allowance}. Requirements: - the caller must have allowance for ``accounts``'s tokens of at least `amount`.

Name	Type	Description
account	address	
amount	uint256	

Returns:

No parameters

checkpoints

Get the `pos`-th checkpoint for `account`.

Name	Type	Description
account	address	
pos	uint32	

Returns:

Name	Type	Description
	tuple	

decimals

Returns the number of decimals used to get its user representation. For example, if `decimals` equals `2`, a balance of `505` tokens should be displayed to a user as `5.05` ($505 / 10^{**2}$). Tokens usually opt for a value of 18, imitating the relationship between Ether and Wei. This is the value {ERC20} uses, unless this function is overridden; NOTE: This information is only used for `_display_` purposes: it in no way affects any of the arithmetic of the contract, including {IERC20-balanceOf} and {IERC20-transfer}.

No parameters

Returns:

Name	Type	Description
	uint8	

decreaseAllowance

Atomically decreases the allowance granted to `spender` by the caller. This is an alternative to {approve} that can be used as a mitigation for problems described in {IERC20-approve}. Emits an {Approval} event indicating the updated allowance. Requirements: - `spender` cannot be the zero address. - `spender` must have allowance for the caller of at least `subtractedValue`.

Name	Type	Description
spender	address	
subtractedValue	uint256	

Returns:

Name	Type	Description
	bool	

delegate

Delegate votes from the sender to `delegatee`.

Name	Type	Description
delegatee	address	

Returns:

No parameters

delegateBySig

Delegates votes from signer to `delegatee`

Name	Type	Description
delegatee	address	
nonce	uint256	
expiry	uint256	
v	uint8	
r	bytes32	
s	bytes32	

Returns:

No parameters

delegates

Get the address `account` is currently delegating to.

Name	Type	Description
account	address	

Returns:

Name	Type	Description
	address	

getPastTotalSupply

Retrieve the `totalSupply` at the end of `blockNumber`. Note, this value is the sum of all balances. It is but NOT the sum of all the delegated votes! Requirements: - `blockNumber` must have been already mined

Name	Type	Description
blockNumber	uint256	

Returns:

Name	Type	Description
	uint256	

getPastVotes

Retrieve the number of votes for `account` at the end of `blockNumber`. Requirements: - `blockNumber` must have been already mined

Name	Type	Description
account	address	
blockNumber	uint256	

Returns:

Name	Type	Description
	uint256	

getRoleAdmin

Returns the admin role that controls `role`. See {grantRole} and {revokeRole}. To change a role's admin, use {_setRoleAdmin}.

Name	Type	Description
role	bytes32	

Returns:

Name	Type	Description
	bytes32	

getVotes

Gets the current votes balance for `account`

Name	Type	Description
account	address	

Returns:

Name	Type	Description
	uint256	

grantRole

Grants `role` to `account`. If `account` had not been already granted `role`, emits a {RoleGranted} event. Requirements: - the caller must have ``role``'s admin role.

Name	Type	Description
role	bytes32	
account	address	

Returns:

No parameters

hasRole

Returns `true` if `account` has been granted `role`.

Name	Type	Description
role	bytes32	
account	address	

Returns:

Name	Type	Description
	bool	

increaseAllowance

Atomically increases the allowance granted to `spender` by the caller. This is an alternative to {approve} that can be used as a mitigation for problems described in {IERC20-approve}. Emits an {Approval} event indicating the updated allowance. Requirements: - `spender` cannot be the zero address.

Name	Type	Description
spender	address	
addedValue	uint256	

Returns:

Name	Type	Description
	bool	

mint

****Add Documentation for the method here****

Name	Type	Description
to	address	
amount	uint256	

Returns:

No parameters

name

Returns the name of the token.

No parameters

Returns:

Name	Type	Description
	string	

nonces

See {IERC20Permit-nonces}.

Name	Type	Description
owner	address	

Returns:

Name	Type	Description
	uint256	

numCheckpoints

Get number of checkpoints for `account`.

Name	Type	Description
account	address	

Returns:

Name	Type	Description
	uint32	

owner

Returns the address of the current owner.

No parameters

Returns:

Name	Type	Description
	address	

pause

Returns true if the contract is paused, and false otherwise.

No parameters

Returns:

No parameters

paused

Returns true if the contract is paused, and false otherwise.

No parameters

Returns:

Name	Type	Description
	bool	

permit

See {IERC20Permit-permit}.

Name	Type	Description
owner	address	
spender	address	
value	uint256	
deadline	uint256	
v	uint8	
r	bytes32	
s	bytes32	

Returns:

No parameters

renounceOwnership

Leaves the contract without owner. It will not be possible to call `onlyOwner` functions anymore. Can only be called by the current owner. NOTE: Renouncing ownership will leave the contract without an owner, thereby removing any functionality that is only available to the owner.

No parameters

Returns:

No parameters

renounceRole

Revokes `role` from the calling account. Roles are often managed via {grantRole} and {revokeRole}: this function's purpose is to provide a mechanism for accounts to lose their privileges if they are compromised (such as when a trusted device is misplaced). If the calling account had been revoked `role`, emits a {RoleRevoked} event. Requirements: - the caller must be `account`.

Name	Type	Description
role	bytes32	
account	address	

Returns:
No parameters

revokeRole

Revokes `role` from `account`. If `account` had been granted `role`, emits a {RoleRevoked} event. Requirements: - the caller must have ``role``'s admin role.

Name	Type	Description
role	bytes32	
account	address	

Returns:
No parameters

supportsInterface

See {IERC165-supportsInterface}.

Name	Type	Description
interfaceId	bytes4	

Returns:

Name	Type	Description
	bool	

symbol

Returns the symbol of the token, usually a shorter version of the name.
No parameters

Returns:

Name	Type	Description
	string	

totalSupply

See {IERC20-totalSupply}.
No parameters

Returns:

Name	Type	Description
	uint256	

transfer

See {IERC20-transfer}. Requirements: - `recipient` cannot be the zero address. - the caller must have a balance of at least `amount`.

Name	Type	Description
recipient	address	
amount	uint256	

Returns:

Name	Type	Description
	bool	

transferFrom

See {IERC20-transferFrom}. Emits an {Approval} event indicating the updated allowance. This is not required by the EIP. See the note at the beginning of {ERC20}. Requirements: - `sender` and `recipient` cannot be the zero address. - `sender` must have a balance of at least `amount`. - the caller must have allowance for ``sender``'s tokens of at least `amount`.

Name	Type	Description
------	------	-------------

Name	Type	Description
sender	address	
recipient	address	
amount	uint256	

Returns:

Name	Type	Description
	bool	

transferOwnership

Transfers ownership of the contract to a new account (``newOwner``). Can only be called by the current owner.

Name	Type	Description
newOwner	address	

Returns:

No parameters

unpause

****Add Documentation for the method here****

No parameters

Returns:

No parameters