

- Excepciones
 - Mecanismo de tratamiento y recuperación de errores en tiempo de ejecución
 - Capturar (detectar) una excepción
 - Manejar la excepción
 - Cuando se produce una excepción
 - se suspende la secuencia de ejecución normal
 - se busca un manejador de la excepción

- Excepciones

- Ejemplo

```
<?php
echo "Hola Mundo\n";
try {
    .....
    .....
    ... excepción
    // El código siguiente no es ejecutado.
    echo 'No se ejecuta';
}
catch (Exception $e) {
    echo 'Excepción atrapada: ', $e->getMessage(), "\n";
}

// Continúa la ejecución
echo "Hola Mundo\n";
?>
```

- Excepciones
 - Las excepciones son objetos de la clase Exception

```
class Exception {
    protected $message = 'Unknown exception'; // exception message
    protected $code = 0;                      // user defined exception code
    protected $file;                          // source filename of exception
    protected $line;                          // source line of exception

    function __construct($message = null, $code = 0);

    final function getMessage();              // message of exception
    final function getCode();                 // code of exception
    final function getFile();                 // source filename
    final function getLine();                 // source line
    final function getTrace();                // an array of the backtrace()
    final function getTraceAsString();        // formatted string of trace

    /* Overrideable */
    function __toString();                    // formatted string for display
}
```

- Excepciones

- Las excepciones pueden ser lanzadas por programa

```
<?php
try {
    $error = 'Always throw this error';
    throw new Exception($error);

    // Code following an exception is not executed.
    echo 'Never executed';
}
catch (Exception $e) {
    echo 'Caught exception: ', $e->getMessage(), "\n";
}

// Continue execution
echo 'Hello World';
?>
```

- Excepciones
 - El usuario puede definir sus propias excepciones extendiendo la clase Exception

```
class MyException extends Exception {  
    // Redefine the exception so message isn't optional  
    public function __construct($message, $code = 0) {  
        // some code  
        // make sure everything is assigned properly  
        parent::__construct($message, $code);  
    }  
  
    // custom string representation of object  
    public function __toString() {  
        return __CLASS__ . ": [{$this->code}]: {$this->message}\n";  
    }  
  
    public function customFunction() {  
        echo "A Custom function for this type of exception\n";  
    }  
}
```

- Excepciones

```
class TestException {
    public $var;
    const THROW_NONE      = 0;
    const THROW_CUSTOM    = 1;
    const THROW_DEFAULT   = 2;

    function __construct($avalue = self::THROW_NONE) {
        switch ($avalue) {
            case self::THROW_CUSTOM:           // throw custom exception
                throw new MyException('1 is an invalid parameter', 5);
                break;
            case self::THROW_DEFAULT:          // throw default one.
                throw new Exception('2 isnt allowed as a parameter', 6);
                break;
            default:                           // No exception, object will be created.
                $this->var = $avalue;
                break;
        }
    }
}
```

Qué pasa con la excepciones lanzadas acá?
➔ propagación

- Excepciones

```
// Example 1
try {
    $o = new TestException(TestException::THROW_CUSTOM);
}
catch (MyException $e) {          // Will be caught
    echo "Caught my exception\n", $e;
    $e->customFunction();
}
catch (Exception $e) {           // Skipped
    echo "Caught Default Exception\n", $e;
}

// Continue execution
var_dump($o);
echo "\n\n";
```

- Excepciones

```
// Example 2
try {
    $o = new TestException(TestException::THROW_DEFAULT);
}
catch (MyException $e) {          // Doesn't match this type
    echo "Caught my exception\n", $e;
    $e->customFunction();
}
catch (Exception $e) {           // Will be caught
    echo "Caught Default Exception\n", $e;
}

// Continue execution
var_dump($o);
echo "\n\n";
```


- Excepciones

```
// Example 3
try {
    $o = new TestException(TestException::THROW_CUSTOM);
}
catch (Exception $e) {           // Will be caught
    echo "Default Exception caught\n", $e;
}

// Continue execution
var_dump($o);
echo "\n\n";
```

- Excepciones

```
// Example 4
try {
    $o = new TestException();
}
catch (Exception $e) {           // Skipped, no exception
    echo "Default Exception caught\n", $e;
}

// Continue execution
var_dump($o);
echo "\n\n";
?>
```