

Typowe domknięcia: Ajax i zdarzenia

Domknięcia najczęściej spotykamy w wywołaniach zwrotnych:

```
function lookupStudentRecord(studentID) {
    ajax(
        'https://some.api/student/${ studentID }',
        function onRecord(record) {
            console.log(
                '${ record.name } (${ studentID })'
            );
        }
    );
}

lookupStudentRecord(114);
// Frank (114)
```

Wywołanie zwrotne `onRecord(..)` zostanie wywołane w przyszłości, gdy wróci odpowiedź z wywołania Ajax. To wywołanie zostanie zainicjalizowane z wnętrza narzędzia `ajax(..)`, skądkolwiek ono pochodzi. Co więcej, nastąpi to długo po wykonaniu `lookupStudentRecord(..)`.

Jak to możliwe, że zmienna `studentID` będzie nadal dostępna w wywołaniu zwrotnym? Odpowiedzią jest domknięcie.

Domknięcia są też często wykorzystywane do obsługi zdarzeń:

```
function listenForClicks(btn, label) {
    btn.addEventListener("click", function onClick(){
        console.log(
            'Przycisk ${ label } został kliknięty!'
        );
    });
}

var submitBtn = document.getElementById("submit-btn");

listenForClicks(submitBtn, "Checkout");
```

Wywołanie zwrotne obsługi zdarzenia `onClick(..)` domyka się na parametrze `label`. Gdy przycisk zostanie kliknięty, zmienna `label` będzie nadal istniała i będzie można jej użyć. Jest to domknięcie.